

Grado en Ingeniería Electrónica Industrial y Automática
Departamento de Ingeniería de Sistemas y Automática



**Universidad
Carlos III de Madrid**
www.uc3m.es

TRABAJO DE FIN DE GRADO

“PROYECTO AT-AS”

*Diseño de un robot cuadrúpedo
imprimible, fabricación e implementación
de la locomoción.*

Autor:
Tutor:

Ana de Prado Navarrete
Alberto Valero Gómez

Universidad Carlos III de Madrid
Septiembre de 2012

AGRADECIMIENTOS

Antes que nada quiero dedicar este proyecto a los mejores padres del mundo. Siempre presentes, han sido para mí la mejor guía que he podido recibir en mi vida.

En primer lugar, quiero hacer un especial reconocimiento al profesor Alberto Valero Gómez, mi tutor, por su disposición en todo momento para ayudarme y aconsejarme, por los ánimos recibidos y por permitirme realizar este proyecto que reviste mi especial interés.

También agradezco su aportación a Juan González Gómez quien junto con Alberto me embarcaron en el mundo de las impresoras 3D y reavivaron mi motivación por desarrollar proyectos robóticos.

Mi más profundo agradecimiento a mis compañeros de equipo: María Ramos Montero, Carlos Rodríguez Zambrana y Jose Ignacio Manzano Fraile, con quienes compartí innumerables momentos fuera y dentro de los laboratorios. Sin ellos no habría disfrutado tanto la carrera.

Por último, quiero agradecer a todos mis amigos y familiares por su infinito apoyo y preocupación, en especial a tres fantásticas personas: Alejandro González y mis hermanos María de Prado y Javier de Prado.

¡Gracias!



RESÚMEN

El presente proyecto consiste en el diseño, fabricación y control del robot cuadrúpedo imprimible llamado ‘AT-AS’. Ha sido realizado en el departamento de Ingeniería de Sistemas y Automática de la Escuela Politécnica Superior de la Universidad Carlos III de Madrid como Trabajo de fin de Grado de la titulación de Grado en Ingeniería Electrónica Industrial y Automática.

A continuación se citan los objetivos que se han cumplido para la creación y desarrollo del proyecto AT-AS:

- **Estudio de la cinemática y aplicaciones de los distintos robots ‘n-podos’** (robots andadores de ‘n’ patas) para la elección del modelo físico óptimo sobre el que se ha basado el desarrollo de todo el proyecto.
- **Diseño del robot cuadrúpedo AT-AS.**
- **Impresión de las piezas diseñadas** en plástico ABS gracias a impresoras 3D.
- **Montaje de la estructura física del robot e instalación de la electrónica** necesaria para su control.
- **Control remoto por medio de un *Gamepad*** conectado al PC.
- **Comunicación PC-robot inalámbrica** a través de módulos *Bluetooth* y módulos *Xbee*.
- **Implementación y evaluación del diseño.** Programación de movimientos básicos.

El AT-AS es un proyecto de **robot eficiente**, porque con pocos recursos técnicos y económicos se ha conseguido una base útil y completa de robot andador. Además, es un proyecto de **robot flexible** ya que su diseño está abierto a muchas posibilidades de evolución para trabajos futuros: como la mutación del modelo cuadrúpedo a un modelo hexápodo, la adaptabilidad para invertir el plano de avance, la facilidad para el cambio en el diseño de las patas y estructura modular, la posibilidad de adición de sensores para mejora del control o la comunicación con otros robots por medio de redes inalámbricas gracias a los módulos Xbee.

ABSTRACT

The present project consists of the design, construction and control of printable quadruped robot called 'AT-AS'. It has been carried out at the System Engineering and Automation Department at Carlos III Polytechnic School University in Madrid as a Final Project for the Bachelor in Automation and Industrial Electronics Engineering.

The following objectives have been accomplished for the creation and development of the AT-AS:

- **Kinematics study and applications of the different ‘*n-pod*’ robots** (*n-legged* walking robots) for the optimal choice of the physical model on which the development of the entire project is based.
- **Design of the AT-AS quadruped robot.**
- **Printing of the designed pieces** in ABS plastic using 3D printers.
- **Assembly of the robot's physical structure and installation of the necessary electronic components** for its control.
- **Remote control through a Gamepad** connected to a PC.
- **PC-robot wireless communication** through *Bluetooth* and *XBee* modules.
- **Design implementation and evaluation.** Basic movement programming.

The AT-AS is a project of an **efficient robot**, as it has achieved a useful and complete walking robot base using few technical and economic resources. In addition, it is a **flexible robot** project since its design is open to many development possibilities in future works such as the transformation of the quadruped model to a hexapod model, the adaptability to invert the advancement plane, easy modification of leg and modular structure design, the possibility of adding sensors to improve control or communication with other robots using wireless networks via *Xbee* modules.

ÍNDICE:

CAPÍTULO 1: INTRODUCCIÓN	10
1.1. INTRODUCCIÓN.....	10
1.2. MARCO DEL PROYECTO Y ANTECEDENTES	12
1.3. OBJETIVOS DEL PROYECTO.....	13
1.4. FASES DEL PROYECTO	14
1.5. ELEMENTOS Y MEDIOS UTILIZADOS.....	15
CAPÍTULO 2: COMPONENTES DEL PROYECTO	17
2.1. IMPRESORA 3D.....	17
2.2. SERVOMOTORES	19
2.3. ARDUINO MEGA 2560	21
2.4. BATERÍA LIPO Y UBEC	23
2.5. CONTROL REMOTO: GAMEPAD.....	25
2.6. CONEXIÓN INALÁMBRICA: MÓDULOS BLUETOOTH Y XBEE	26
CAPÍTULO 3: FASES ESPECÍFICAS PARA EL DISEÑO DEL AT-AS	29
3.1. ESTUDIO DE LOS DIFERENTES ROBOTS N-PODOS.....	29
3.2. ELECCIÓN DEL MODELO FÍSICO DEL AT-AS	39
3.3. ESTUDIO CINEMÁTICO Y DE LOCOMOCIÓN	40
3.4. DISEÑO DE LAS PIEZAS DE LA ESTRUCTURA IMPRIMIBLE	46
CAPÍTULO 4: FABRICACIÓN E IMPLEMENTACIÓN	51
4.1. FABRICACIÓN Y MONTAJE.....	51
4.2. ENVÍO DE DATOS Y CONTROL REMOTO.....	66
4.3. IMPLEMENTACIÓN FINAL Y EVALUACIÓN DEL AT-AS.	71
CAPÍTULO 5: CONCLUSIÓN	77
CAPÍTULO 6: BIBLIOGRAFÍA Y REFERENCIAS	81
ANEXOS	82
ANEXO I: PRESUPUESTO Y TIEMPO EMPLEADO.....	82
ANEXO II: PLANOS DE LAS PIEZAS DISEÑADAS.....	85

ÍNDICE DE FIGURAS:

<i>Figura 1 - Timberjack Walking Machine, hexápodo de uso forestal</i>	10
<i>Figura 2 - Robot ATHLETE desarrollado por ingenieros de la NASA</i>	11
<i>Figura 3 - Robots imprimibles desarrollados por alumnos de la UC3M</i>	12
<i>Figura 4 - MADRE impresora 3D</i>	18
<i>Figura 5 - UC3PO impresora 3D</i>	18
<i>Figura 6 - GANDALF impresora 3D</i>	19
<i>Figura 7 - Servomotores</i>	20
<i>Figura 8 - Servo motor Futaba S3003</i>	21
<i>Figura 9 - Arduino Mega 2560</i>	22
<i>Figura 10 - Baterías LiPo</i>	23
<i>Figura 11 - Batería Lipo DragonRed 7,4V - 5000mAh - 30C</i>	24
<i>Figura 12 - Cargador imaxRC 3B para baterías LiPo de 2S y 3S</i>	24
<i>Figura 13 - UBEC HobbyWing 5-6V</i>	25
<i>Figura 14 - GamePAD FireStorm Dual Analog 3 de Thrustmaster</i>	25
<i>Figura 15 - Módulo Bluetooth</i>	26
<i>Figura 16 - Vistas acotadas de un módulo Xbee</i>	27
<i>Figura 17 - Módulo Xbee S1 y módulo Xbee con antena</i>	27
<i>Figura 18 - Xbee Explorer USB</i>	28
<i>Figura 19 - Xbee Shield compatible con Arduino</i>	28
<i>Figura 20 - Artrópodos</i>	29
<i>Figura 21 - Tetrápodos</i>	29
<i>Figura 22 - Robot ASTERISK empujando un obstáculo y escalando una malla</i>	30
<i>Figura 23 - Robot ASTERISK manipulando un objeto</i>	30
<i>Figura 24 - Robot saltador simple y robot saltador de Toyota</i>	32
<i>Figura 25 - Robots bípedos</i>	33
<i>Figura 26 - Robot REX para personas con problemas de movilidad reducida</i>	33
<i>Figura 27 - Robots cuadrúpedos</i>	34
<i>Figura 28 - Robots hexápodos</i>	35
<i>Figura 29 - Robot hexápodo levantando una pata sin problemas de estabilidad</i>	36
<i>Figura 30 - Eslabones y articulaciones</i>	37
<i>Figura 31 - Grados de libertad en las patas</i>	37
<i>Figura 32 - Movimiento vertical de robot con pata de tres grados de libertad</i>	38
<i>Figura 33 - Movimiento no-vertical de robot con pata de dos grados de libertad</i>	38
<i>Figura 34 - Cuadrúpedo AT-AS con patas de 2 grados de libertad</i>	39
<i>Figura 35 - Esquema cinemático con ejes de coordenadas</i>	40
<i>Figura 36 - Esquema cinemático de una pata</i>	40
<i>Figura 37 - Esquema cinemático con ángulos y ejes de coordenadas</i>	41
<i>Figura 38 - Distancias entre los ejes de giro L_1 y L_2</i>	42
<i>Figura 39 - Ángulos de giro máximos</i>	43
<i>Figura 40 - Caballo caminando avanza en modo ‘pata por pata’</i>	44
<i>Figura 41 - Centro de Gravedad avance pata por pata</i>	44

<i>Figura 42 - Caballo trotando avanza en modo ‘alternando par’</i>	45
<i>Figura 43 - Centro de Gravedad avance ‘alternando par’</i>	45
<i>Figura 44 - Piezas imprimibles del AT-AS</i>	46
<i>Figura 45 - Esquema de los nombres de las piezas</i>	46
<i>Figura 46 - Diseño del ‘pie’ del cuadrúpedo con hueco para acoplar el servo</i>	47
<i>Figura 47 - Diseño de uno de los ‘huesos’ y de un ‘botón’</i>	48
<i>Figura 48 - Diseño de la pieza ‘cadera’ que sujeta las piernas al esqueleto central</i>	49
<i>Figura 49 - Diseño de la ‘columna central’</i>	50
<i>Figura 50 - Diseño del eje central cuadrado</i>	50
<i>Figura 51 - Piezas impresas en plástico ABS</i>	51
<i>Figura 52 - Diagrama de montaje del esqueleto central</i>	52
<i>Figura 53 - Detalle de los agujeros para tornillo de las caderas</i>	52
<i>Figura 54 - Diagrama de montaje de los servos en las ‘caderas’ que sujetan las patas</i>	53
<i>Figura 55 - Diagrama de montaje del servo dentro del ‘pie’</i>	54
<i>Figura 56 - Diagrama de montaje de los ‘huesos’ entre sí con la posición de los botones</i>	54
<i>Figura 57 - Fijación de los discos de los servomotores</i>	55
<i>Figura 58 - Montaje de la articulación del pie. Detalle de cómo pegar el botón</i>	55
<i>Figura 59 - Detalle de cómo debe quedar el botón</i>	56
<i>Figura 60 - Pierna montada al completo</i>	56
<i>Figura 61 - Vista del montaje completo de todas las piezas impresas junto a los servos</i>	57
<i>Figura 62 - Detalle del orificio para tornillo del eje cuadrado central</i>	58
<i>Figura 63 - Tres tiras de velcro abrazando la batería Lipo</i>	58
<i>Figura 64 - Dos tiras de velcro largas</i>	59
<i>Figura 65 - Detalle de velcro en la parte baja</i>	59
<i>Figura 66 - Detalle de velcro en la parte alta</i>	60
<i>Figura 67 - Montaje final de la batería LiPo</i>	60
<i>Figura 68 - Esquema de conexión de la alimentación del AT-AS</i>	61
<i>Figura 69 - Foto de los cables y conectores eléctricos</i>	62
<i>Figura 70 - Circuito alimentación y control servos</i>	62
<i>Figura 71 - Vista de los pines largos del mini circuito</i>	63
<i>Figura 72 - Conexión mini circuito con salidas digitales de Arduino</i>	63
<i>Figura 73 - Numeración de los servos</i>	64
<i>Figura 74 - Conexión del módulo Bluetooth</i>	65
<i>Figura 75 - Instalación Xbee Shield en Arduino Mega 2560</i>	66
<i>Figura 76 - Esquema de comunicaciones entre elementos del proyecto</i>	66
<i>Figura 77 - Esquema general del programa C++</i>	67
<i>Figura 78 - Esquema general del programa Arduino</i>	69
<i>Figura 79 - Botones y Joysticks del Gamepad</i>	71
<i>Figura 80 - Arco izquierdo y derecho</i>	73
<i>Figura 81 - Goma elástica recortada en tira de 5cm</i>	75
<i>Figura 82 - Detalle de las gomas de los pies</i>	76
<i>Figura 83 - Evolución a hexápodo del diseño AT-AS</i>	78
<i>Figura 84 - Diseño de las piezas ‘cadera lateral’ y ‘cadera central’</i>	79
<i>Figura 85 - Adaptabilidad del AT-AS si se invierte el plano de avance</i>	79

ÍNDICE DE TABLAS:

<i>Tabla 1 - Nombre del robot según el número de patas.....</i>	<i>31</i>
<i>Tabla 2 - Composición de cada pierna</i>	<i>47</i>
<i>Tabla 3 - Composición del esqueleto</i>	<i>49</i>
<i>Tabla 4 - Conexión de Servos con salidas digitales Arduino</i>	<i>64</i>
<i>Tabla 5 - Letra correspondiente a cada servo</i>	<i>70</i>
<i>Tabla 6 - Botones pulsados del Gamepad</i>	<i>72</i>

CAPÍTULO 1: INTRODUCCIÓN

1.1. INTRODUCCIÓN

Un robot es un dispositivo electrónico y generalmente mecánico capaz de desempeñar tareas automáticamente, ya sea por medio de supervisión humana directa, a través de un programa predefinido o siguiendo un conjunto de reglas generales establecidas. Existen multitud de robots diseñados para cumplir diferentes objetivos: como brazos robóticos para el montaje de piezas en una fábrica, vehículos aéreos no tripulados o los robots móviles terrestres.

Muchas de las investigaciones en la robótica de hoy en día se enfocan hacia el desarrollo de máquinas caminantes. Esto es debido a que en algunas aplicaciones se requiere una considerable movilidad, sobre todo cuando se tiene la necesidad de locomoción en terrenos irregulares.

El vehículo hexápodo de uso forestal de la *Figura 1* llamado “*Timberjack Walking Machine*” (1) es un claro ejemplo de la facilidad de los robots con patas de poder transitar por lugares inaccesibles para vehículos sobre ruedas e incluso sobre orugas.



Figura 1 - Timberjack Walking Machine, hexápodo de uso forestal

Existen diferentes configuraciones en cuanto al tipo de desplazamiento que llevan a cabo los robots móviles. Las dos formas de locomoción más comunes son el uso de ruedas y el uso de patas. Es en este punto donde se plantea el primer problema o cuestión a resolver del actual análisis de robots móviles terrestres: ¿rueda o pata?

¿Rueda o pata?

La movilidad y maniobrabilidad de un sistema de patas es mayor que el de uno con ruedas, pero lo complicado de su control y diseño hace que su uso hoy en día esté restringido en casi todos los casos a robótica experimental.

Sin embargo, millones de años de evolución de la naturaleza terrestre que dieron como resultado estructuras en forma de pata demuestran que se trata de un eficaz medio de locomoción. Si bien tiene desventajas en comparación con otros sistemas inventados por el hombre, a lo largo de esta memoria veremos que sigue teniendo ventajas muy importantes que no debemos despreciar.

¿Acaso un robot con ruedas puede saltar un edificio? ¿Puede avanzar en terreno con muchos obstáculos? ¿Se puede manipular objetos con las ruedas? Estas son algunas de las preguntas que los ingenieros se plantean y que llevan a conclusiones donde el uso de la rueda tiene limitaciones serias.

Una opción muy acertada para solucionar la duda entre rueda o pata es utilizar las ventajas de ambos métodos de locomoción, fusionar ambos diseños añadiendo ruedas a los pies de unas piernas robóticas, como es el caso de ‘ATHLETE’ (Figura 2), robot andador hexápodo con ruedas desarrollado por ingenieros de la NASA (2).



Figura 2 - Robot ATHLETE desarrollado por ingenieros de la NASA

En este Trabajo de fin de Grado se plantea desarrollar la creación de un robot con patas imprimible.

1.2. MARCO DEL PROYECTO Y ANTECEDENTES

Este proyecto ha sido desarrollado en el departamento de Ingeniería de Sistemas y Automática de la Escuela Politécnica Superior de la Universidad Carlos III de Madrid como Trabajo de Fin de Grado (TFG) de la titulación de Grado en Ingeniería Electrónica Industrial y Automática.

Forma parte de una serie de proyectos que comienzan con la idea de crear pequeños robots móviles cuyas piezas puedan ser impresas con las impresoras 3D *MakerBot (Thing-O-Matic)* del departamento de Sistemas y Automática de la Universidad.

Un grupo de estudiantes de la Universidad Carlos III bajo la supervisión de los profesores Alberto Valero y Juan González Gómez, aprendieron las artes necesarias para la impresión en 3D, diseñando y construyendo a lo largo de 2011 y 2012 robots móviles imprimibles como los que se muestran en la *Figura 3*.

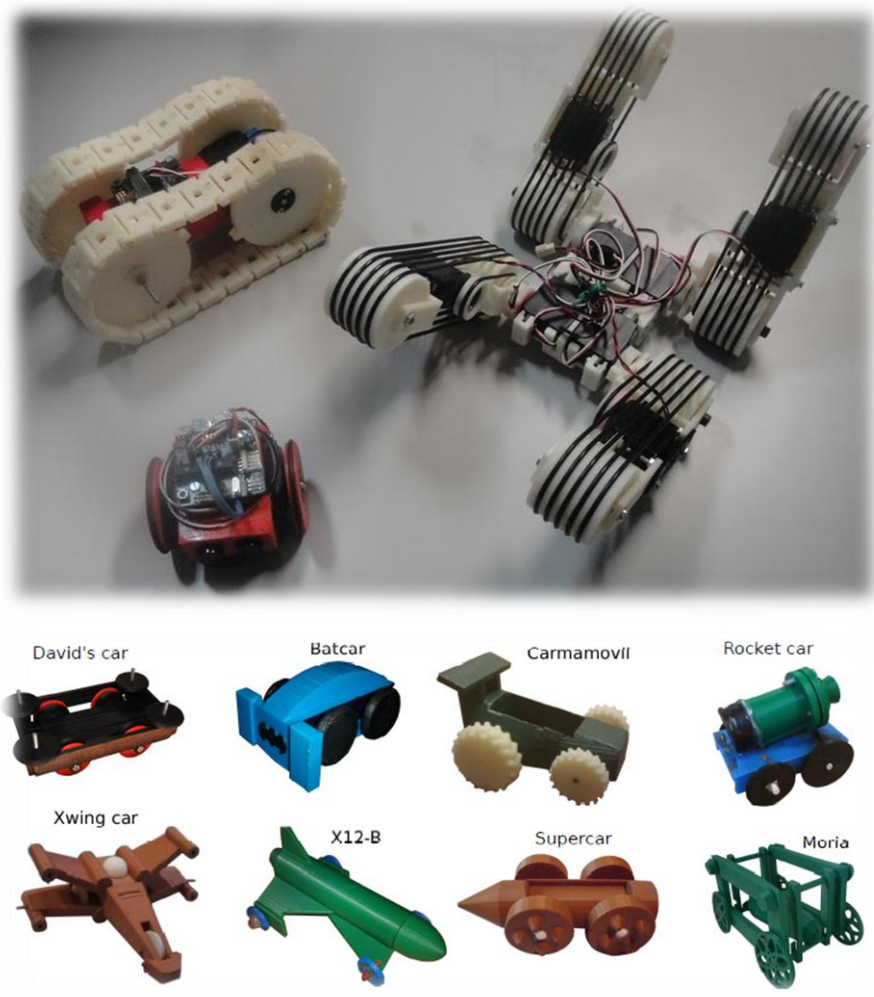


Figura 3 - Robots imprimibles desarrollados por alumnos de la UC3M

1.3. OBJETIVOS DEL PROYECTO

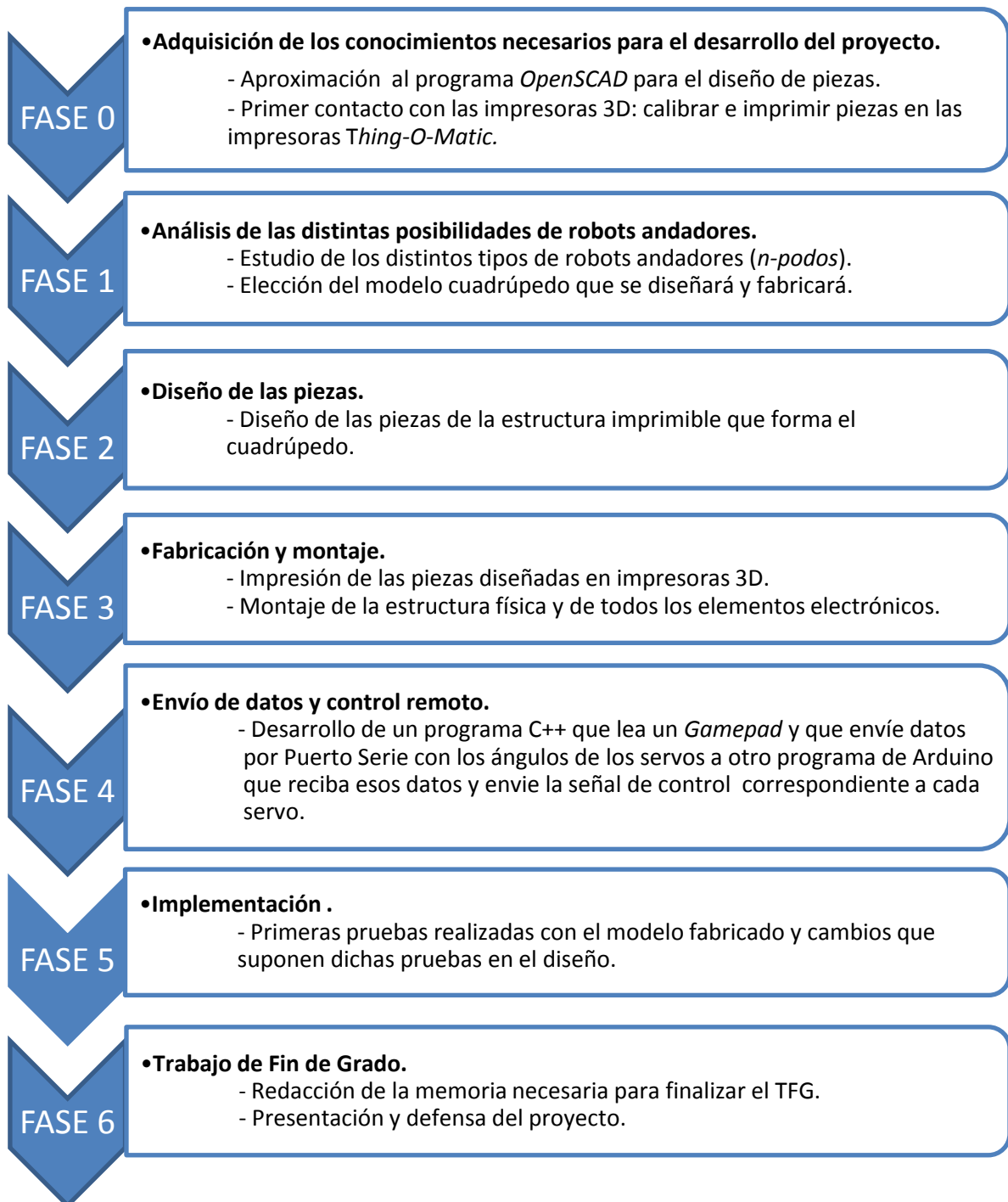
El presente proyecto tiene como meta principal el diseño, construcción y control del robot cuadrúpedo imprimible AT-AS. Para alcanzar dicha meta de manera satisfactoria se ha hecho necesaria la demarcación de otros objetivos en base a los contenidos específicos de cada una de las fases del proyecto.

Se enumera a continuación la sucesión de objetivos que se han buscado para dar forma al proceso de creación de AT-AS:

- I. **Estudio de la cinemática y aplicaciones de los distintos robots ‘n-podos’** (robots andadores de ‘n’ patas) para la elección del modelo físico óptimo sobre el que se ha basado el desarrollo de todo el proyecto.
- II. **Diseño del robot cuadrúpedo AT-AS.**
- III. **Impresión de las piezas diseñadas** en plástico ABS gracias a impresoras 3D *Thing-O-Matic*.
- IV. **Montaje de la estructura física del robot e instalación de la electrónica** necesaria para su control.
- V. **Control remoto por medio de un GamePAD** conectado al PC.
- VI. **Comunicación PC-robot inalámbrica** a través de módulos Bluetooth y protocolo *ZigBee* (módulos Xbee).
- VII. **Implementación y evaluación del diseño.** Programación de movimientos básicos.

1.4. FASES DEL PROYECTO

A continuación se muestra un resumen de las fases generales seguidas para la elaboración del proyecto de diseño y construcción del AT-AS que ha conformado este Trabajo de Fin de Grado:



1.5. ELEMENTOS Y MEDIOS UTILIZADOS

En el siguiente listado se nombran todos los componentes utilizados para la realización del proyecto de fabricación del cuadrúpedo imprimible AT-AS:

- **Impresoras 3D:** ‘MADRE’ y ‘UC3PO’, ambas modelos *Thing-O-Matic* de la empresa *Makerbot*, disponibles en el departamento de Sistemas y Automática de la Universidad. Con ellas se imprimieron en plástico ABS todas las piezas que forman el esqueleto del AT-AS, a excepción de la *pieza central* que al ser demasiado larga para dichas impresoras se imprimió en ‘GANDALF’, un modelo *RepRap* perteneciente al profesor Alberto Valero.
- **Servomotores Futaba modelo S3003:** ha sido necesario el uso de 8 servos en total para la locomoción del AT-AS, dos por cada una de las cuatro patas del cuadrúpedo.
- **Arduino Mega 2560:** Microcontrolador basado en *ATmega2560*.
- **Batería LiPo de doble celda 7.4V (2S), 5000mAh y 30C:** empleada para alimentar todos los elementos electrónicos del proyecto.
- **Cargador específico de baterías LiPo de doble celda:** puesto que no es posible cargarla por medio de otros cargadores convencionales.
- **Regulador de voltaje UBEC (5-6V):** para transformar los 7.4V de la batería LiPo en 5V y evitar el exceso de tensión que dañaría los servomotores.
- **Elementos de comunicación inalámbrica.** Se han probado dos sistemas distintos de comunicación inalámbrica por Puerto Serie:
 - a) Mediante un **módulo Bluetooth**.
 - b) Mediante dos **módulos Xbee** (Serie 1), con sus correspondientes adaptadores *Shield* para Arduino y adaptador *Explorer USB* para la conexión con el ordenador.
- **Gamepad:** periférico sencillo usado para el control remoto en el cual se ha basado la programación de movimientos según los *joysticks* y botones pulsados por el usuario.
- **Ordenador portátil** con sistema operativo **Ubuntu** que gracias a un programa C++, ha servido como centro de control calculando las posiciones que los servos tienen en cada momento y enviando dicha información a la placa Arduino a través de módulos inalámbricos. Además, se han instalado en el

ordenador los programas y librerías necesarios para el diseño e impresión de las piezas:

- **OpenSCAD:** software para la creación de objetos *CAD* (del inglés ‘*Computer-Aided Design*’) sólidos en 3D.
 - **Biblioteca OOML** (*Object Oriented Mechanics Library*): librería C++ que permite realizar diseños de objetos en 3D y luego generar el código OpenSCAD. (2)
 - **Qt Creator:** Compilador C++.
 - **Arduino 0019:** Entorno de código abierto para la programación de la placa *Arduino Mega 2560*.
 - **ReplicatorG:** necesario para imprimir los diseños desde los archivos en formato ‘.stl’
-
- **Otros materiales y herramientas para el montaje y conexiones del AT-AS:** tales como tornillos, tuercas, tiras de velcro, cable conductor, gomas elásticas, soldador, estaño, placa perforada para circuitos, tira de pines, destornilladores, mini-taladro y pegamento instantáneo.

En el siguiente capítulo, se va a desarrollar con más detalle algunos de los elementos importantes de la lista anterior.

CAPÍTULO 2: COMPONENTES DEL PROYECTO

2.1. IMPRESORA 3D

En ingeniería los diseños de piezas en 3D creados se plasman primero en planos y finalmente se transforman en objetos reales. Este proceso de llevar las ideas del plano a la realidad es lento y requiere disponer de los materiales y maquinaria adecuados.

Las impresoras 3D son máquinas capaces de construir objetos físicos a partir de sus especificaciones digitales.



Se denominan “impresoras” por su similitud con estos dispositivos. Si bien las impresoras 2D producen documentos en papel con texto y gráficos a partir de ficheros digitales mediante la deposición de tinta, las impresoras 3D generan objetos físicos diseñados a partir de la deposición de otros materiales (como plástico y resinas).

En el 2005, el matemático e Ingeniero Adrian Boywer, profesor de la Universidad de Bath en Reino Unido comenzó el proyecto *Reprap*, consistente en la construcción de una impresora 3D capaz de auto-replicarse, es decir, que pueda imprimir sus propias piezas para así crear réplicas de muy bajo coste.

En la actualidad el proyecto *Reprap* ha creado ya la segunda generación de impresoras (*Reprap Mendel*) y ha aparecido todo un ecosistema de microempresas que comercializan los materiales para que los usuarios puedan construirse esta primera “*impresora madre*” con capacidad de auto-replicarse.

Debido a su bajo coste (en torno a los 1000€) muchos usuarios las tienen en sus casas, y ha comenzado lo que han denominado como la “revolución de la fabricación personal”. Miles de personas por todo el mundo están diseñando objetos digitales que pueden ser materializados con estas máquinas como picaportes de puertas, soportes para teléfonos, carcasas, juguetes y pequeños robots. Todo un mundo de objetos que están siendo creados, contruidos y compartidos por toda la comunidad internacional.

A continuación se describen las tres impresoras 3D utilizadas para llevar a cabo el desarrollo del proyecto AT-AS.

MADRE

MADRE es la primera impresora 3D de la Asociación de Robótica de la UC3M (ASROB). El impulsor fue el profesor Juan González. Se encargó de realizar la propuesta, conseguir el dinero, gestionar la compra y ayudar en el montaje y calibración. Se compró con un presupuesto 1200€. A continuación puede observarse una imagen (*Figura 4*) de *MADRE* junto con sus características sacada de la web de ASROB (3):

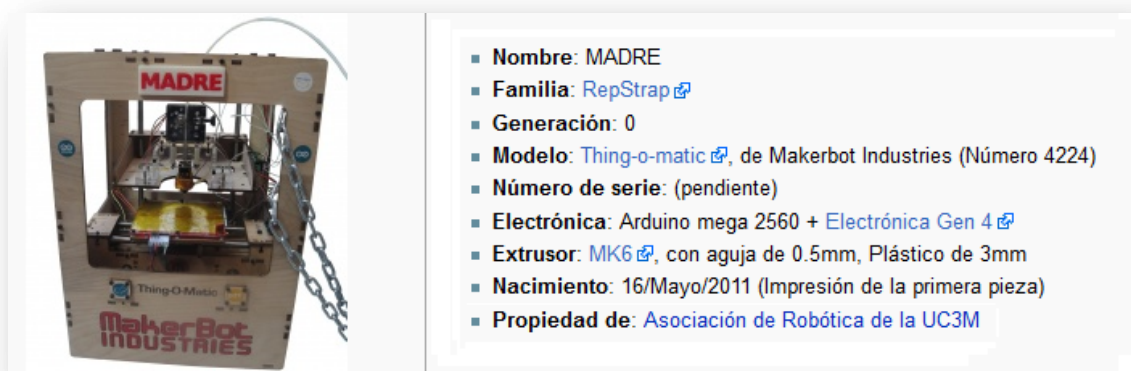


Figura 4 - MADRE impresora 3D

UC3PO

UC3PO, también llamada *PADRE*, es la primera impresora 3D *open-source* comprada por el Departamento de Ingeniería de Sistemas y Automática de la Universidad Carlos III de Madrid (UC3M). El profesor Alberto Valero se encargó de solicitar el dinero, gestionar la compra y ayudar en la construcción y calibración, siendo así el impulsor de la adquisición de dicha impresora. En la siguiente imagen (*Figura 5*) puede verse una foto de la impresora 3D y sus características principales.



Figura 5 - UC3PO impresora 3D

GANDALF the Transparent

La última de las impresoras 3D utilizada para la impresión de las piezas de este proyecto, es ‘GANDALF’ (*Gandalf the Transparent*), un *Clon* de las anteriormente nombradas *PADRE* y *MADRE* perteneciente al profesor Alberto Valero quien la fabricó y calibró. En la *Figura 6* se puede ver una foto y las características principales de la impresora 3D.

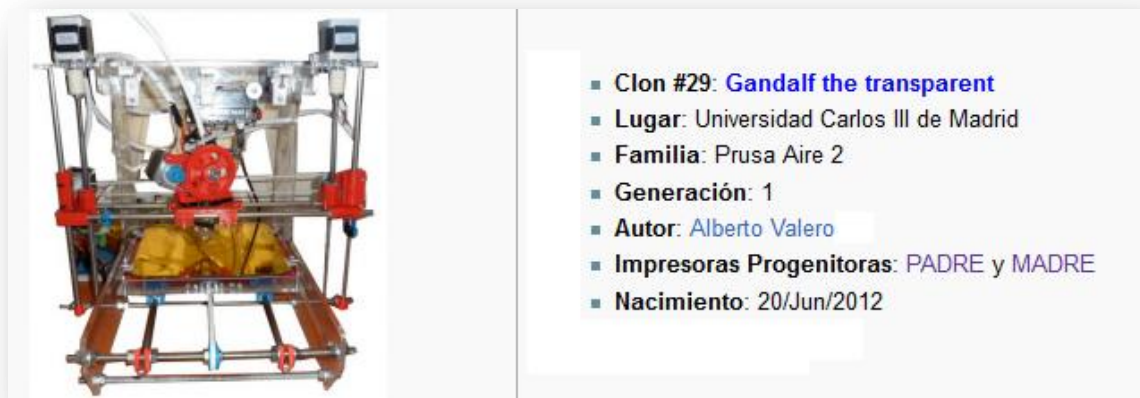


Figura 6 - GANDALF impresora 3D

2.2. SERVOMOTORES

Un *Servo* es un dispositivo pequeño que tiene un eje de posición controlada. El eje del servomotor puede ser llevado a posiciones angulares específicas al transmitirle una señal codificada. Mientras esa señal no varía en la línea de entrada, el servo mantiene la posición angular del engranaje. Cuando la señal codificada cambia, la posición angular de los piñones cambia. En la práctica, se usan servos para posicionar superficies de control como el movimiento de palancas, en timones de juguetes radio control, *animatronics*, y por supuesto en robots.

La principal ventaja de un servomotor sobre los motores tradicionales de corriente continua o alterna, es la incorporación en los mismos de un sistema de retroalimentación. Esta información puede ser utilizada para detectar un movimiento no deseado o para garantizar la exactitud del movimiento ordenado. Un sistema de control de motor que responde a este diseño es conocido como sistema de “Lazo Cerrado”.

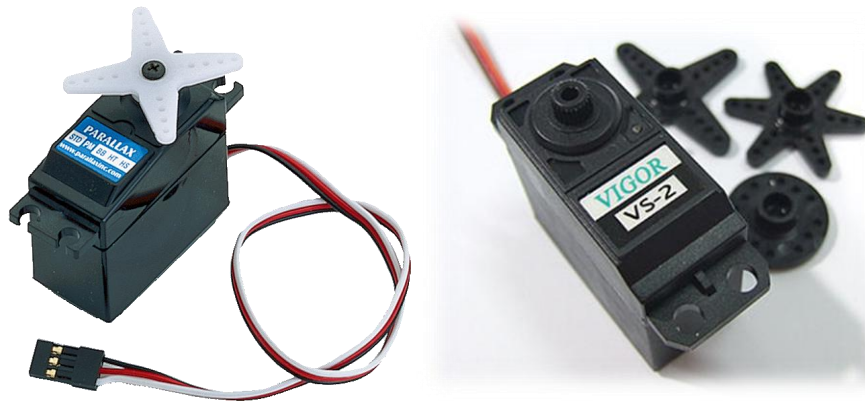


Figura 7 - Servomotores

Los servomotores poseen tres líneas de conexión de entrada:

- Terminal positivo: Recibe la alimentación del motor (4 a 8 voltios).
- Terminal negativo: Referencia tierra del motor (0 voltios).
- Terminal de Control: Recibe la señal de control del motor.

Los colores del cable de cada terminal varían con cada fabricante: el cable del terminal positivo siempre es rojo; el del terminal negativo puede ser marrón o negro; y el del terminal de entrada de señal suele ser de color blanco, naranja o amarillo. El principio de funcionamiento es el mismo, independientemente del modelo e incluso del fabricante.

La señal codificada de control se genera mediante modulación por anchura de pulso, también llamada PWM (*‘Pulse Width Modulation’*). Este sistema consiste en generar una onda cuadrada en la que se varía, dentro de un mismo periodo, el tiempo durante el cual el pulso está a nivel alto con el objetivo de modificar la posición del servo según se desee.

La relación entre posición en grados (α) del eje del servomotor y el ancho de pulso de la señal PWM en milisegundos (t) viene dada por la siguiente expresión:

$$t = 1 + \frac{\alpha}{180}$$

En este proyecto se han utilizado 8 servomotores de la marca *Futaba* modelo *S3003* como el que muestra la siguiente *Figura 8*. Los *Futaba S3003* se alimentan a 5V, y para posicionar el servo es necesario aplicar una señal periódica de 50Hz (20ms de periodo).



Figura 8 - Servo motor Futaba S3003

Las principales características de este modelo son las siguientes:

Dimensiones y peso:

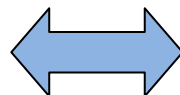
Longitud : 40.3mm
 Anchura : 19.8 mm
 Altura : 36.1 mm
 Peso : 44.4. gr.

Torque o par del motor:

3.2 Kg / 4.8V.
 4.1 Kg / 6V.

2.3. ARDUINO MEGA 2560

Las tarjetas Arduino son plataformas electrónicas de hardware libre. Se componen de una sencilla placa con un microcontrolador que permite desarrollar múltiples prototipos y aplicaciones. Son placas potentes y a la vez muy económicas, aptas para todo tipo de usuarios.



Arduino Mega 2560 es una de las versiones de la tarjeta original de Arduino, basada en el microprocesador *Atmega2560*.

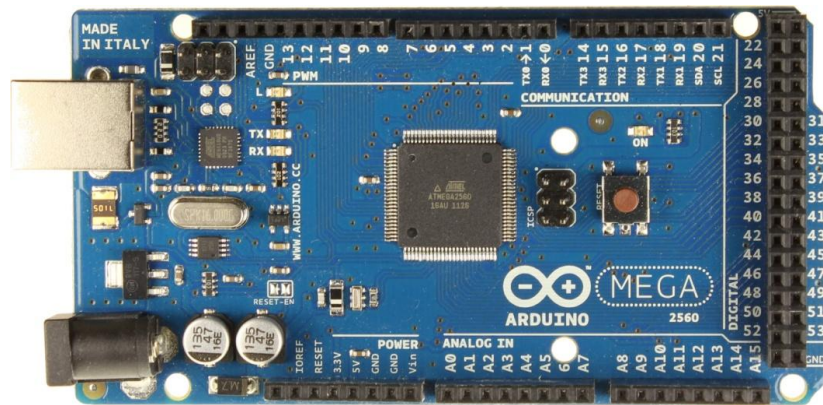


Figura 9 - Arduino Mega 2560

Principales características:

- Tensión de alimentación recomendada: 7-12 V.
- Microprocesador *ATmega2560*.
- Voltaje operativo de 5V.
- 54 Pines Digitales I/O (Entrada/Salida) de las cuales 14 se pueden utilizar como salidas PWM.
- 16 Entradas Analógicas.
- 4 Puertos Serial (UARTs).
- Salida de alimentación de 5V y 3.3V con 50mA.
- Máxima corriente continua para las entradas: 40mA.
- Memoria de programa (*Flash Memory*) de 256kb (el *bootloader* ocupa 8kb).
- Memoria SRAM de 8kb para datos y variables del programa.
- Memoria EEPROM para datos y variables no volátiles de 4Kb.
- Velocidad de trabajo del reloj: 16MHz.
- Dimensiones: 100 x 50 mm.
- Conexión USB.

En este proyecto, se eligió la tarjeta del modelo *Arduino Mega 2560* principalmente por su capacidad, al poder utilizar 14 de los pines digitales como salidas PWM para controlar los servomotores de forma sencilla y eficiente gracias a la librería “*Servo.h*” (3) del propio software de Arduino. Además, esta placa permite la comunicación vía Bluetooth o mediante módulos Xbee con el ordenador gracias a sus puertos *serial (UART)*.

Otra de las razones por la que se seleccionó este modelo es la cantidad de salidas/entradas analógicas y digitales que permiten la futura expansión del proyecto, abriendo múltiples posibilidades como la utilización de sensores de infrarrojos o de ultrasonidos para detectar obstáculos, o la adición de sensores de presión en las terminaciones de las patas para mejorar el control de locomoción ante desniveles e irregularidades del terreno.

2.4. BATERÍA LIPO Y UBEC

Batería LiPo

Las baterías de polímeros de litio, también llamadas ‘LiPo’ (*Lithium Polymer*), son una evolución de las baterías Li-Ion (*Iones de litio*) en las que se ha sustituido el electrolito líquido orgánico por un compuesto sólido, abaratando así costes de producción. Se fabrican en formato rectangular en lugar de cilíndrico, por lo que al construir un pack de baterías de este tipo se ahorra mucho espacio ya que se reduce el hueco entre batería y batería. Además, tienen un peso considerablemente menor al de las baterías basadas en níquel, otro de los modelos de batería más usados del mercado, lo que supone para las LiPo un mejor rendimiento en aplicaciones de aeromodelismo donde su uso se ha extendido considerablemente.



Figura 10 - Baterías LiPo

Las baterías LiPo suelen llevar un identificador alfanumérico que identifica el número de celdas que la componen. Se suele utilizar un número y la letra "S" para identificar el número de celdas conectadas en serie. Cada celda aporta 3,7 V de manera que si una LiPo es de dos celdas en serie "2S" equivale a 7,4V en total. La capacidad se mide en miliamperios/hora (mAh), de forma que si una batería es de "5000mAh" indica que el pack tiene una capacidad de suministrar 5.000 miliamperios (5 amperios) durante una hora completa. Otro identificador de las baterías LiPo es la unidad "C", que nos indica la máxima capacidad de descarga.

Para este proyecto se ha utilizado la **batería LiPo DragonRed de 2S, 5000mAh, 30C y 235g** de peso. En la *Figura 11* puede verse una imagen de la misma. Elegimos usar una batería LiPo y no otros modelos por las siguientes razones:

- Alta densidad de energía, casi el doble que las basadas en níquel.
- Alto voltaje por célula, lo que permite llegar a 7,4 V en dos celdas sin ocupar excesivo volumen y con un reducido peso.
- Poca resistencia interna que permite aprovechar casi el 100% de la energía almacenada.



Figura 11 - Batería Lipo DragonRed 7,4V - 5000mAh - 30C

Cargador Batería LiPo

Las baterías LiPo deben ser cargadas mediante cargadores específicos, por lo que en este proyecto se ha utilizado un **cargador especial ImaxRC B3** apto para baterías LiPo de 2 y 3 celdas (7,4V y 11.1V).



Figura 12 - Cargador imaxRC 3B para baterías LiPo de 2S y 3S

UBEC: Regulador de voltaje para alimentar los servos

Un UBEC es un regulador de voltaje de tipo conmutado ('switching'). Soporta una intensidad máxima de entrada de 5A y es adaptable a voltajes de entrada entre 5,5V y 23V, permitiendo acoplarse a las baterías LiPo de 2S a 5S.

El modelo elegido para este proyecto es el **UBEC de HobbyWing** con voltaje de salida seleccionable (5 ó 6 voltios) como el de la Figura 13. Los 5V de salida que ofrece permiten alimentar los servomotores sin riesgo de daño, ya que no podríamos conectarlos directamente a la batería LiPo de 7,4V.

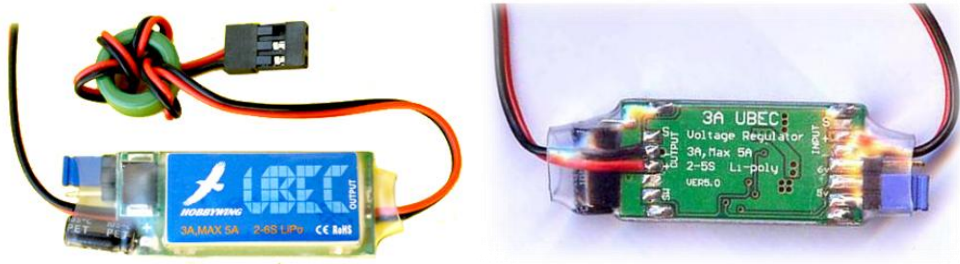


Figura 13 - UBEC HobbyWing 5-6V

2.5. CONTROL REMOTO: GAMEPAD

Un *Gamepad* es un periférico de entrada, un mando apto para todos los usuarios ya que es fácil de usar y comúnmente conocido gracias a la influencia de los videojuegos.

Para el control remoto del AT-AS, se ha elegido el **Gamepad FireStorm Dual Analog 3** de la empresa **Thrustmaster**, el cual se compone de 12 botones en total, entre los que se encuentran dos *joystick* y una cruz multidireccional. Es un mando ergonómico y compacto que incluye todas las características principales de los *Gamepad* del mercado. Se ha elegido este modelo principalmente por su reducido precio.



Figura 14 - GamePAD FireStorm Dual Analog 3 de Thrustmaster

2.6. CONEXIÓN INALÁMBRICA: MÓDULOS BLUETOOTH Y XBEE

Uno de los objetivos de este proyecto es conseguir una comunicación inalámbrica entre el PC y el robot. Se busca eliminar cables y conectores de forma que el AT-AS pueda ser autónomo y desplazarse libremente alejándose del centro de control. En el desarrollo del proyecto se han probado dos métodos distintos de comunicación inalámbrica: los módulos *Bluetooth* y los módulos *Xbee*.

Se pueden utilizar ambos métodos inalámbricos para controlar el AT-AS, sin embargo, los módulos *Xbee* han dado mejores resultados en las primeras pruebas de implementación del robot.

Módulo Bluetooth

Bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de datos entre diferentes dispositivos mediante un enlace por radiofrecuencia, siendo un protocolo de comunicación diseñado especialmente para dispositivos de bajo consumo.

El módulo *Bluetooth* utilizado en el proyecto es el **modelo JY-MCU**. Dispone de 4 pines como vemos en la *Figura 15*, dos de alimentación (*VCC* y *GND*) y otros dos de comunicación (*RXD* y *TXD*).

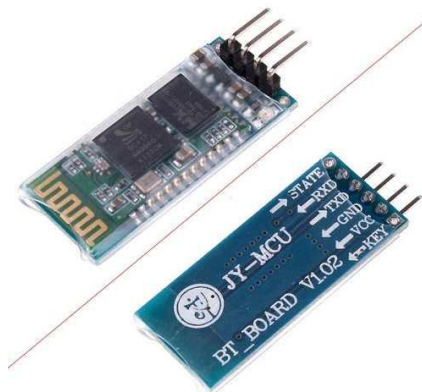


Figura 15 - Módulo Bluetooth

Las principales ventajas de este módulo son su facilidad de conexión y su reducido coste, ya que está basado en transceptores de bajo costo. Su gran desventaja es la distancia que abarca. Al instalarlo en el AT-AS, se ha comprobado que la cobertura es de sólo unos pocos metros antes de dar problemas de comunicación. Por este motivo, se probaron también los módulos *Xbee* que mejoraban notablemente el alcance de la comunicación inalámbrica.

Módulo Xbee

Los módulos *XBee* están basados en el protocolo de comunicación IEEE 802.15.4, también conocido como ‘*ZigBee*’. Tienen múltiples aplicaciones, como la de implementar redes de sensores y actuadores, pero en nuestro caso van a ser utilizados para crear una comunicación inalámbrica entre el AT-AS y el ordenador. Para realizar dicha comunicación, se necesitan dos módulos *Xbee*: uno que servirá de emisor desde el PC y otro que funcionará como receptor conectado a la placa Arduino.

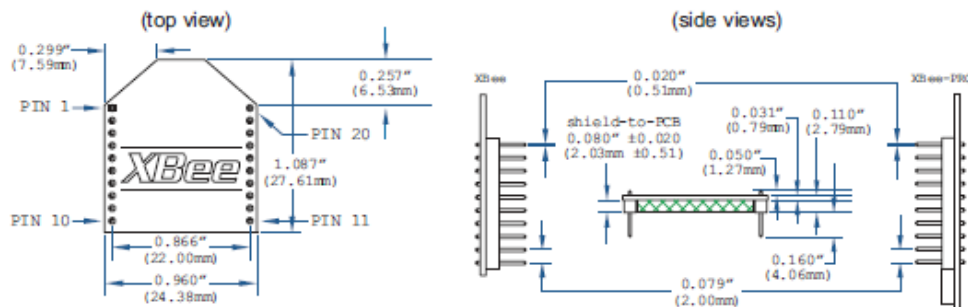


Figura 16 - Vistas acotadas de un módulo Xbee

Se ha elegido para este proyecto los módulos **Xbee de Serie 1** con antena, como los que se muestran en la Figura 17.

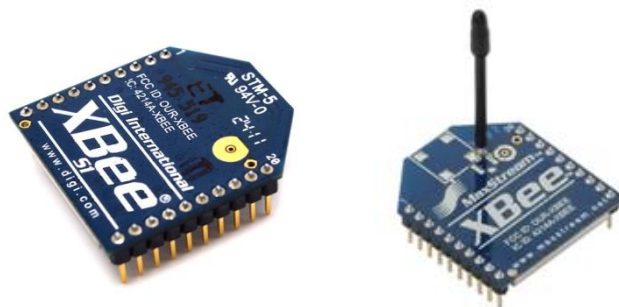


Figura 17 - Módulo Xbee S1 y módulo Xbee con antena

Algunas de las características de estos módulos:

- Bajo consumo: menos de 50mA cuando están en funcionamiento y 10 μ A en modo ‘sleep’.
- Posibilidad de tener muchos dispositivos en una misma red: 65.000 direcciones para cada uno de los 16 canales disponibles.
- Interfaz serial.
- Mayor alcance que el Bluetooth: la comunicación entre módulos *Xbee* llega hasta los 100m en exteriores y 30m en interiores.

Debido a que los módulos *Xbee* tienen una separación de pines de 2mm y no como cualquier *protoboard* estándar con separación de 0.1 pulgadas entre pines, debemos utilizar adaptadores especiales para su conexión. Para poder conectar uno de los módulos *Xbee* con el ordenador, es necesario utilizar un adaptador USB como el ***Xbee Explorer USB*** de la *Figura 18*.



Figura 18 - Xbee Explorer USB

Para conectar el segundo de los módulos *Xbee* al *Arduino Mega 2560*, se ha utilizado una placa adaptadora ***Xbee Shield*** compatible con varios de los modelos de *Arduino* como la que se muestra a continuación en la *Figura 19*.



Figura 19 - Xbee Shield compatible con Arduino

CAPÍTULO 3: FASES ESPECÍFICAS PARA EL DISEÑO DEL AT-AS

3.1. ESTUDIO DE LOS DIFERENTES ROBOTS N-PODOS

Artrópodos y Tetrápodos. El origen e inspiración de los robots andadores.

Comúnmente se conoce como ‘patas’ a las extremidades o apéndices de un animal que sostienen su cuerpo y permiten su traslado por tierra. Se trata de estructuras pares y articuladas. Estrictamente hablando, en el reino animal sólo poseen patas los vertebrados tetrápodos y los invertebrados artrópodos.

El término **artrópodo** proviene del griego ‘*Arthropoda*’ (*árthron* «articulación» y *poús* «pie») y se aplica a los animales invertebrados con apéndices articulados, los cuales muestran una gran plasticidad evolutiva y que han dado lugar a las estructuras más diversas como antenas, mandíbulas, quelíceros y por supuesto, las patas.



Figura 20 - Artrópodos

A su vez, el término **tetrápodo** proviene del griego ‘*Tetrapoda*’ («cuatro patas»). Este grupo abarca anfibios, reptiles, aves (con sus antepasados dinosaurios) y todos los mamíferos.

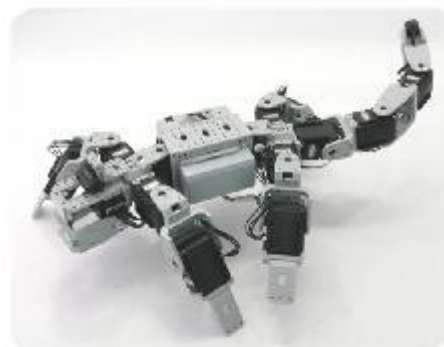


Figura 21 - Tetrápodos

Esta es una clasificación general según el número de extremidades de los seres vivos. Sin embargo, que un animal tenga cuatro extremidades (tetrápodo) no significa que camine con las cuatro (en cuyo caso hablamos de un cuadrúpedo), sino que puede tener cuatro extremidades y utilizar sólo dos de ellas para caminar (bípedo) y las otras dos para otras funciones como es el caso del ser humano.

Para la robótica, la naturaleza ha sido una clara fuente de inspiración a la hora de diseñar los modelos de robots con estructuras articuladas. Además, se suelen buscar funcionalidades distintas en las patas de estos robots, por ejemplo utilizar una misma pata para avanzar y para manipular objetos es una gran ventaja que se puede conseguir añadiendo sensores, ventosas, pinzas y otros elementos.

En las siguientes imágenes (*Figura 22* y *Figura 23*) se pueden ver varias pruebas que realiza ‘*ASTERISK*’ (4), robot hexápodo desarrollado en la Universidad de Osaka. *ASTERISK* es capaz entre otras funciones de trepar, empujar y hasta manipular un objeto con dos de sus patas.

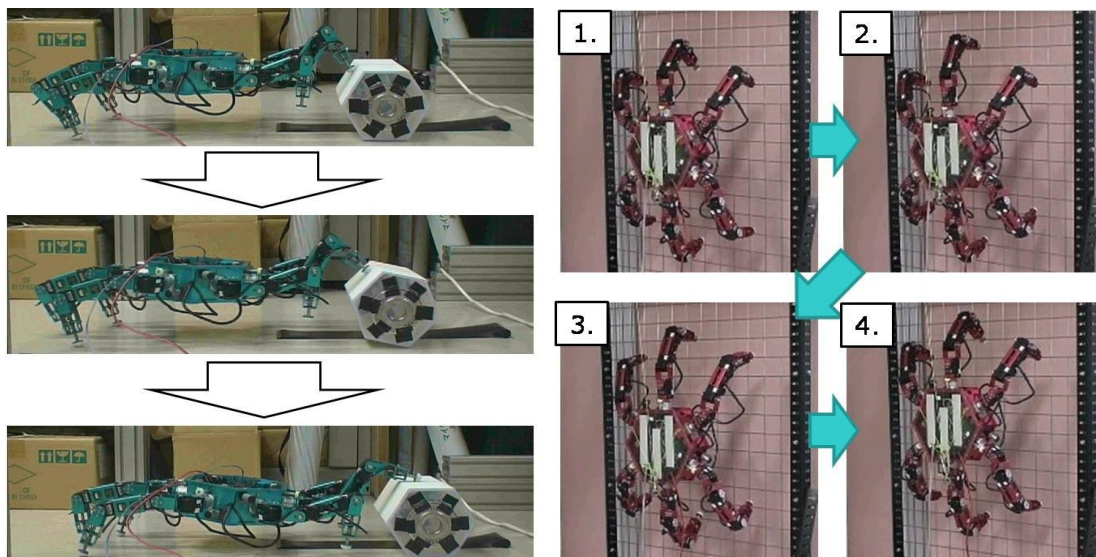


Figura 22 - Robot ASTERISK empujando un obstáculo y escalando una malla

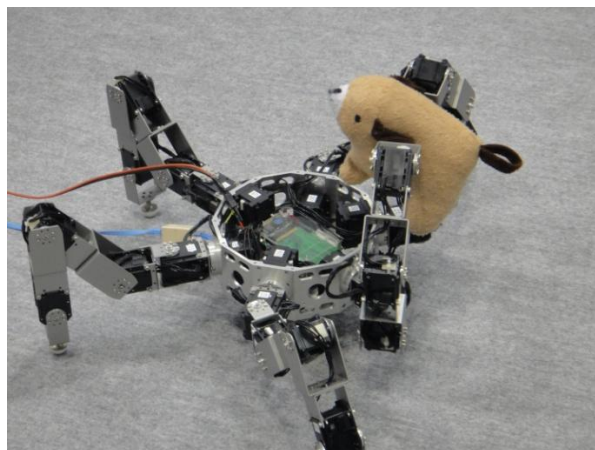


Figura 23 - Robot ASTERISK manipulando un objeto

Robots andadores “n-podos”

Un **robot andador** (*legged robot*) es un robot móvil cuya locomoción se basa en el uso exclusivo de piernas en lugar de ruedas o de orugas.

Como bien dice su nombre, podemos clasificar los robots ‘n-podos’ (*n-patas*) según el número de patas:

Número de patas	Nombre Robot
0	<i>Apodo</i>
1	<i>Saltador (*)</i>
2	<i>Bípedo (*)</i>
3	<i>Trípode</i>
4	<i>Cuadrúpedo (*)</i>
5	<i>Pentápodo</i>
6	<i>Hexápodo (*)</i>
8	<i>Octópodo</i>

Tabla 1 - Nombre del robot según el número de patas

En la tabla anterior [Tabla 1] están señalados con un asterisco (*) los tipos de robots andadores más utilizados en robótica. A continuación se van a describir brevemente cada uno de estos diseños con sus ventajas y desventajas para así entender mejor el modelo elegido y desarrollado en este Trabajo de Fin de Grado.

▪ Una pata – Robots Saltadores

El número mínimo de piernas que un robot necesita para tener contacto con el suelo es una. Estos robots son diseñados generalmente para ser ‘robots de salto’.



Figura 24 - Robot saltador simple y robot saltador de Toyota

Para poder mantenerse en pie, el robot saltador debe saltar todo el tiempo ya que si se detiene se caerá. Estos robots son dinámicamente estables y consiguen mantener su equilibrio cambiando su propio centro de gravedad mediante la aplicación de fuerzas correctivas para evitar la caída cuando se les empuja o desestabiliza.

Ventajas:

La ventaja de tener una sola pata es que pueden dar un salto para evitar obstáculos grandes. Tomando impulso pueden llegar a sobrepasar obstáculos más altos que la altura de su paso normal.

Dado que sólo hay una única ‘pierna’ controlable, los diseñadores pueden ignorar los problemas de coordinación entre extremidades. Además, un robot con una sola pata es más eficiente energéticamente en comparación con los robots que utilizan más patas.

Desventajas:

El inconveniente es su diseño complejo y problemático control de estabilidad sobre todo a la hora de frenar el movimiento de avance a saltos.

▪ *Dos patas – Robots Bípedos*

Desde el comienzo de la robótica, los estudios sobre robots bípedos se llevan a cabo de forma continua, especialmente hacia formas humanoides. El objetivo principal de éstos robots es que lleguen a caminar, subir y bajar escaleras, correr, pararse e incluso bailar.

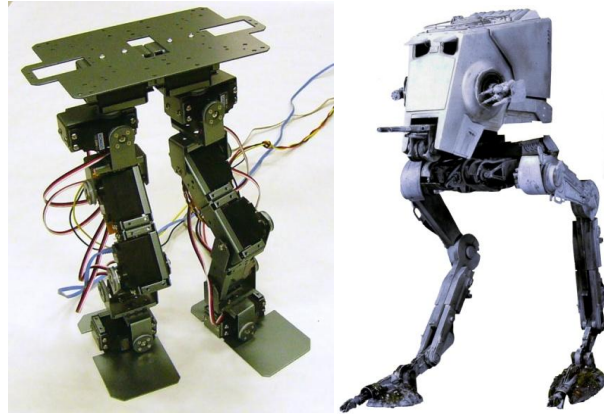


Figura 25 - Robots bípedos

Ventajas:

Al igual que los robots saltadores, su reducido número de patas conlleva menor peso, menor consumo energético y menor dificultad en la coordinación entre las piernas.

Como se ha mencionado anteriormente, se suele buscar la similitud a las formas humanoides. Un robot que se parezca a un ser humano implica que puede ser utilizado en el mismo ambiente de trabajo que un humano sin realizar ningún cambio o adaptación para la implantación del mismo. Además, pueden ser utilizados como ayuda a la movilidad de personas con alguna discapacidad como es el ejemplo del robot ‘REX’ creado en Nueva Zelanda por *RexBionics* (5) con el objetivo de sustituir la silla de ruedas.



Figura 26 - Robot REX para personas con problemas de movilidad reducida

Desventajas:

Para que los robots bípedos sean dinámicamente estables, equilibrarse al ponerse en pie y al avanzar, requieren complejos algoritmos de control. Estos robots deben planificar muy bien sus pasos, controlando la distribución del peso y modificando el centro de gravedad de tal manera que al próximo paso, el robot no se desestabilice.

▪ *Cuatro patas – Robots Cuadrúpedos*

Los robots cuadrúpedos son estáticamente estables al tener cuatro puntos de apoyo con la superficie del suelo. Si se colocan sus patas en la posición y ángulos correctos, son difíciles de tumbar por agentes externos.



Figura 27 - Robots cuadrúpedos

El patrón de marcha de un robot cuadrúpedo se puede diseñar de diferentes formas. Los **modos de avance cuadrúpedo** más básicos son los siguientes:

- Pata por pata: En este modo de avance, hay tres puntos de contacto con la superficie en todo momento. Mover una pata a la vez hace que el robot avance más lento y gaste más energía, pero lo mantiene estable.
- Alternando par: En este otro modo, las patas se mueven de dos en dos. En cualquier momento el robot tiene dos puntos de contacto con la superficie del suelo creando un robot dinámicamente estable. Es un modo más rápido y eficiente, pero menos estable estáticamente que el primer modo ‘pata por pata’.

Ventajas:

Como se mencionaba anteriormente, los cuadrúpedos son estáticamente estables, es decir, cuando están parados no tienen problemas para mantener la posición. Además, el poder levantar una de las patas manteniendo tres puntos de contacto con el suelo y así mantenerse aún estables es una ventaja si se quiere manipular objetos con dicha pata. Otra ventaja más es que los robots de cuatro patas consiguen esquivar más fácilmente obstáculos que los robots bípedos o los saltadores con sus problemas de estabilidad.

Desventajas:

Al tener más elementos, más patas y motores que controlar, requieren complejos algoritmos de control para coordinarse y conseguir una locomoción eficaz. También requieren mayor gasto energético para mover todas sus extremidades.

▪ Seis patas – Robots Hexápodos

Los robots hexápodos tienen características y modos de locomoción similares a los robots de cuatro patas, amplificando las ventajas a la hora de esquivar obstáculos y mantener estabilidad, pero también amplificando las desventajas al añadirse dos patas más que controlar y alimentar energéticamente.

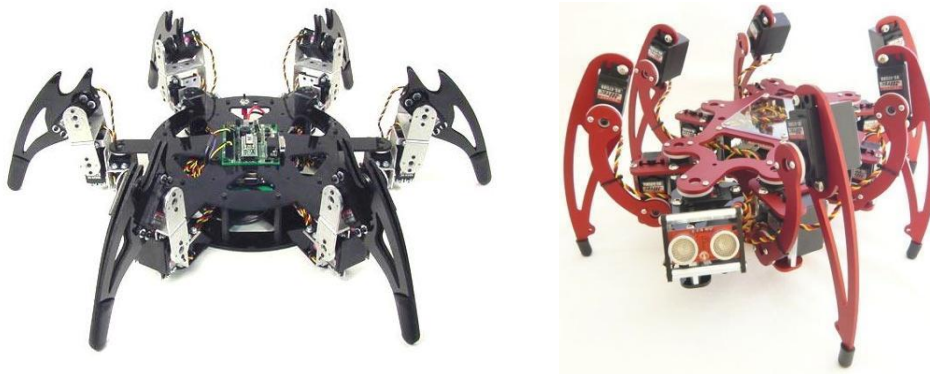
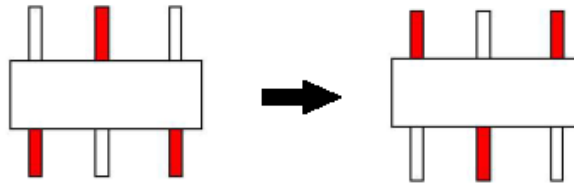


Figura 28 - Robots hexápodos

El patrón de marcha de un robot hexápodo también se puede diseñar de diferentes formas. Los **modos de avance hexápodo** más básicos son los siguientes:

- Pata por pata: Al igual que los robots cuadrúpedos, mover una sola pierna a la vez hace que el robot avance más lento y gaste más energía. El robot mantiene cinco puntos de contacto con la superficie en todo momento lo que le proporciona mucha estabilidad tanto dinámica como estática.

- Alternando trío: Este otro modo es más efectivo que el modo “alternar par” de los cuadrúpedos, ya que el robot hexápodo va alternando de tres en tres patas apoyadas sobre el suelo proporcionándole tanto estabilidad estática como dinámica. El hexápodo puede detener su marcha sin desestabilizarse gracias a que tiene al menos tres patas apoyadas en todo momento.



Ventajas:

Los robots hexápodos son aún más estables estáticamente que los robots cuadrúpedos y el resto de robots con menos patas. Como se explicaba anteriormente, tienen en todo momento varios puntos de apoyo con la superficie. Pueden aprovechar las patas no apoyadas para esquivar obstáculos y manipular objetos sin desestabilizarse.

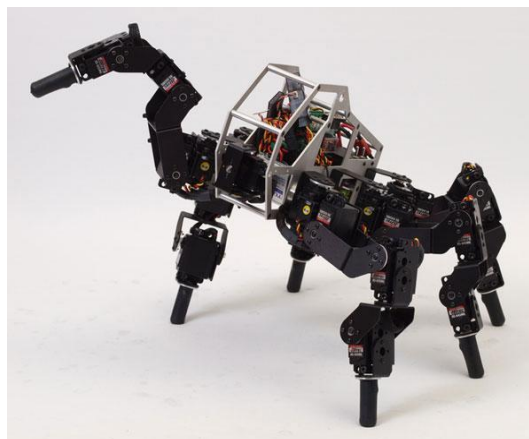


Figura 29 - Robot hexápodo levantando una pata sin problemas de estabilidad

Desventajas:

Los robots con seis patas tienen aún más complicaciones para coordinar sus extremidades que los robots con menos patas. Necesitan mayor aporte energético y algoritmos de control aún más elaborados.

Grados de libertad de las patas según el número de servos

Cada pata de un robot andador está formada por una serie de elementos estructurales sólidos llamados **eslabones**. Los eslabones van unidos entre sí mediante **articulaciones** que permiten un movimiento relativo entre cada dos eslabones consecutivos.

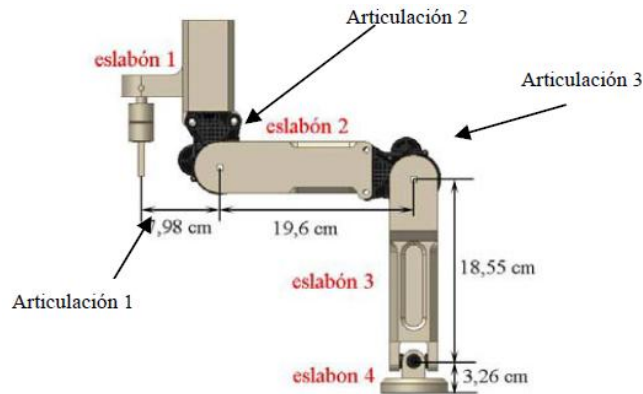
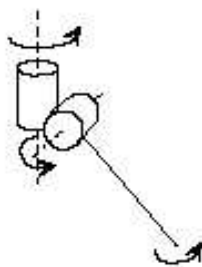


Figura 30 - Eslabones y articulaciones

Cada uno de los movimientos independientes (giros o desplazamientos) que puede realizar cada articulación respecto a la anterior es un **grado de libertad (GDL)**. En el caso de robots andadores que utilizan servomotores, cada servomotor provoca en la articulación un movimiento de giro, luego ‘*n*’ articulaciones con servomotor proveerá a la pata de ‘*n*’ grados de libertad.

Pata con 2 GDL



Pata con 3 GDL

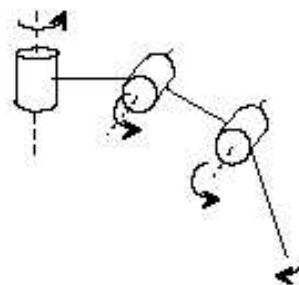


Figura 31 - Grados de libertad en las patas

Se pueden fabricar patas de robots andadores con un solo GDL (es decir, un solo actuador) en adelante. A continuación se exponen las diferencias entre los robots andadores con tres grados de libertad por pata y los que sólo tienen dos grados de libertad por pata, ya que han sido los modelos estudiados y analizados para la creación del AT-AS.

- **Robot andador de 3 GDL por pata:**

Estos robots incluyen tres servomotores en cada una de sus patas, formando hasta un total de tres eslabones y tres articulaciones. Esta configuración es la más utilizada comúnmente en patas articuladas. Una de las ventajas importantes que tiene este modelo es que proporciona buena maniobrabilidad con las patas, posibilitando la manipulación de objetos y ayudando al robot andador a esquivar mejor obstáculos gracias a la facilidad para levantar verticalmente la pata y el cuerpo. En la siguiente imagen (*Figura 32*), se puede apreciar las ventajas del movimiento vertical.

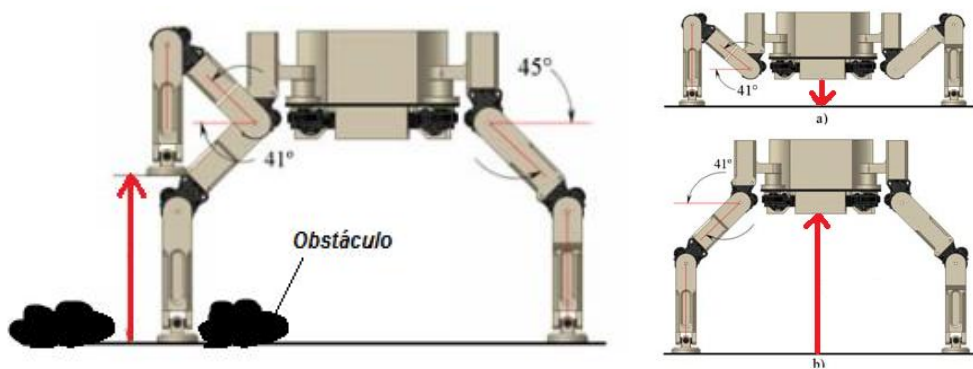


Figura 32 - Movimiento vertical de robot con pata de tres grados de libertad

- **Robot andador de 2 GDL por pata:**

Estos robots andadores tienen dos actuadores por pata formando hasta un total de dos eslabones y dos articulaciones. Esta segunda opción es menos útil que la primera (3 GDL) ya que al robot le resultará más difícil esquivar obstáculos y tendrá su capacidad para manipular objetos reducida a esos dos GDL. Sin embargo, el avance del robot sigue siendo eficaz.

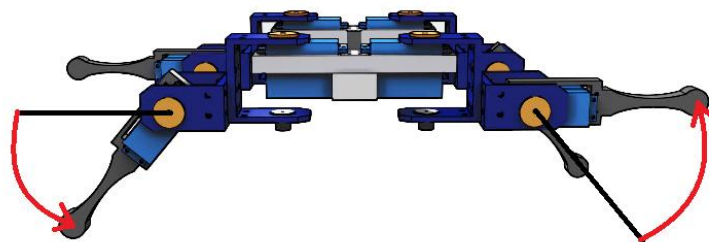


Figura 33 - Movimiento no-vertical de robot con pata de dos grados de libertad

3.2. ELECCIÓN DEL MODELO FÍSICO DEL AT-AS

Una vez analizadas las distintas posibilidades de robots andadores, se ha elegido para desarrollar el diseño del robot que conforma este TFG un modelo **cuadrúpedo de 2 grados de libertad por pata**, es decir, robot andador de cuatro patas y dos servos en cada una de ellas. En la siguiente *Figura 34* se muestran varias vistas del diseño al cual se le ha denominado como ‘AT-AS’ [*All Terrain - Anita Scout*].



Figura 34 - Cuadrúpedo AT-AS con patas de 2 grados de libertad

Como hemos estudiado en los apartados anteriores, la elección de un modelo cuadrúpedo de dos grados de libertad por pata tiene desventajas en comparación con otros *n-podos*: no evita obstáculos tan fácilmente como los robots que incluyen más patas y tampoco puede manipular objetos como lo haría un robot con más grados de libertad en las patas.

¿Por qué entonces se ha elegido este modelo de robot andador? Porque al tener menos piezas que diseñar e imprimir, menos servos y componentes, se obtienen las siguientes ventajas:

- ✓ Más sencillo de diseñar.
- ✓ Más barato de fabricar.
- ✓ Mayor simplicidad de programación.

Y la razón más importante de todas:

- ✓ Aun siendo más simple que otros modelos, es suficiente para tener una **base útil y completa de robot andador**, sobre la cual se podrían llegar a desarrollar fácilmente otros modelos *n-podos* en caso de ampliar el proyecto en un futuro.

3.3. ESTUDIO CINEMÁTICO Y DE LOCOMOCIÓN

Estudio cinemático del AT-AS:

Es preciso estudiar la cinemática de las patas del robot para poder realizar el control de la posición de dichas patas en coordenadas espaciales, es decir, determinar qué posiciones han de tomar las articulaciones para posicionar el extremo de la pata en un punto concreto con respecto al sistema de coordenadas externas del sistema. Éste es un problema similar a los resueltos para el control de los brazos robóticos de los robots manipuladores.

En la siguiente imagen (*Figura 35*), se presenta el esquema cinemático de una pata del cuadrúpedo. Puede observarse que tiene dos grados de libertad, es decir, dos articulaciones con un posible giro cada una. Justo debajo, la *Figura 36* muestra otro esquema cinemático directamente sobre una imagen del diseño de una de las patas.

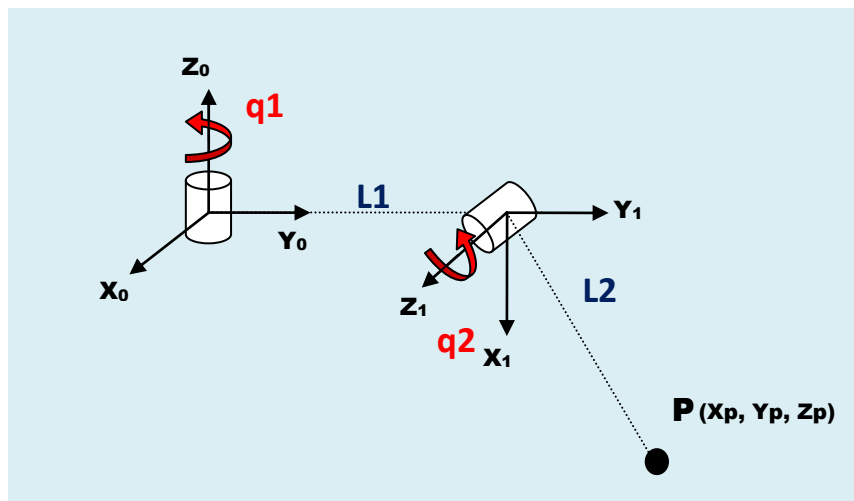


Figura 35 - Esquema cinemático con ejes de coordenadas

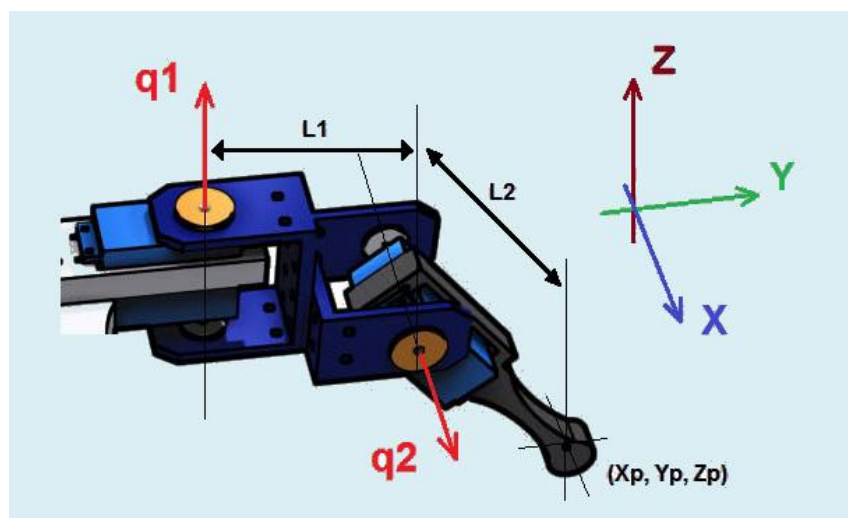


Figura 36 - Esquema cinemático de una pata

El modelo cinemático directo e inverso de una de las patas del AT-AS se ha obtenido mediante **relaciones geométricas**, que no es un método sistemático como la utilización del ‘*Algoritmo de Denavit-Hartenberg*’ (6) pero sí eficiente y sencillo para sistemas con pocos grados de libertad como el que ocupa el presente proyecto con tan sólo dos grados de libertad (dos giros de los servos según los ángulos θ_1 y θ_2).

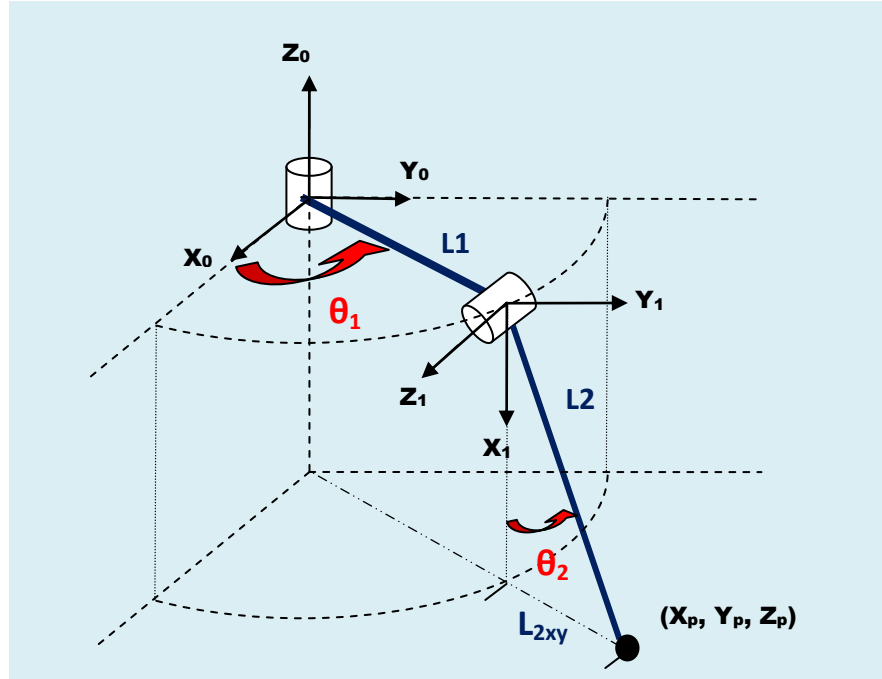


Figura 37 - Esquema cinemático con ángulos y ejes de coordenadas

En la Figura 37, se indican las variables L_1 y L_2 correspondientes a la longitud de los eslabones, así como las variables θ_1 y θ_2 correspondientes a los desplazamientos angulares de cada articulación. Siguiendo las variables anteriores y mediante relaciones geométricas y trigonométricas, se han obtenido las ecuaciones de **cinemática directa de las coordenadas** (X_p, Y_p, Z_p) del punto final ‘P’. En primer lugar calculamos la coordenada X_p :

$$X_p = (L_1 + L_{2xy}) \cdot \cos \theta_1$$

Siendo L_{2xy} , señalada en la Figura 37, una variable auxiliar que representa la proyección de L_2 sobre el plano XY:

$$L_{2xy} = L_2 \cdot \sin \theta_2$$

Sustituyendo los parámetros de dicha variable, la ecuación final de la coordenada X_p queda de la siguiente forma:

$$(1) \quad X_p = (L_1 + L_2 \cdot \sin \theta_2) \cdot \cos \theta_1$$

Siguiendo el mismo procedimiento, se calcula la coordenada Y_p también dependiente de la variable auxiliar L_{2xy} :

$$Y_p = (L_1 + L_{2xy}) \cdot \sin \theta_1 \quad \rightarrow$$

$$(2) \quad Y_p = (L_1 + L_2 \cdot \sin \theta_2) \cdot \sin \theta_1$$

La coordenada Z_p necesita menos parámetros para definirse, tan sólo depende del ángulo θ_2 y de la longitud L_2 . La ecuación es para calcularla es:

$$(3) \quad Z_p = -(L_2 \cdot \cos \theta_2)$$

La **longitud de los eslabones** (L_1 y L_2), es fija y se calculan midiendo las distancias entre los ejes de giro como indica la siguiente *Figura 38*:

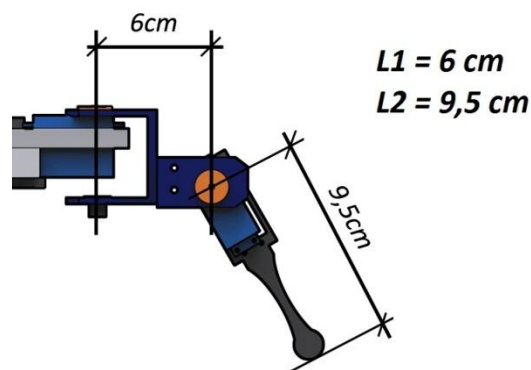


Figura 38 - Distancias entre los ejes de giro L_1 y L_2

Para calcular mediante **cinemática inversa los ángulos de giro** (θ_1 y θ_2), tan sólo hay que despejar de las ecuaciones anteriores dichas variables.

Despejando Ecuación (3):

$$\theta_2 = \cos^{-1} \left(\frac{Z_p}{L_2} \right)$$

Dividiendo Ecuación (2) entre (1):

$$\frac{Y_p}{X_p} = \frac{(L_1 + L_{2xy})}{(L_1 + L_{2xy})} \cdot \tan \theta_1 = \tan \theta_1$$

Por tanto el ángulo que falta:

$$\theta_1 = \tan^{-1}\left(\frac{Y_p}{X_p}\right)$$

Los ángulos de giro (θ_1 y θ_2) son los desplazamientos angulares que sufren las dos articulaciones de la pata. Los rangos máximos que soportan las articulaciones con servos instalados pueden verse en la siguiente *Figura 39*. Como señala dicha imagen, el rango máximo de cada articulación es de 180° .

Rango de movilidad articulaciones ($\theta_{1\text{ máx}}$ y $\theta_{2\text{ máx}}$) = 180°

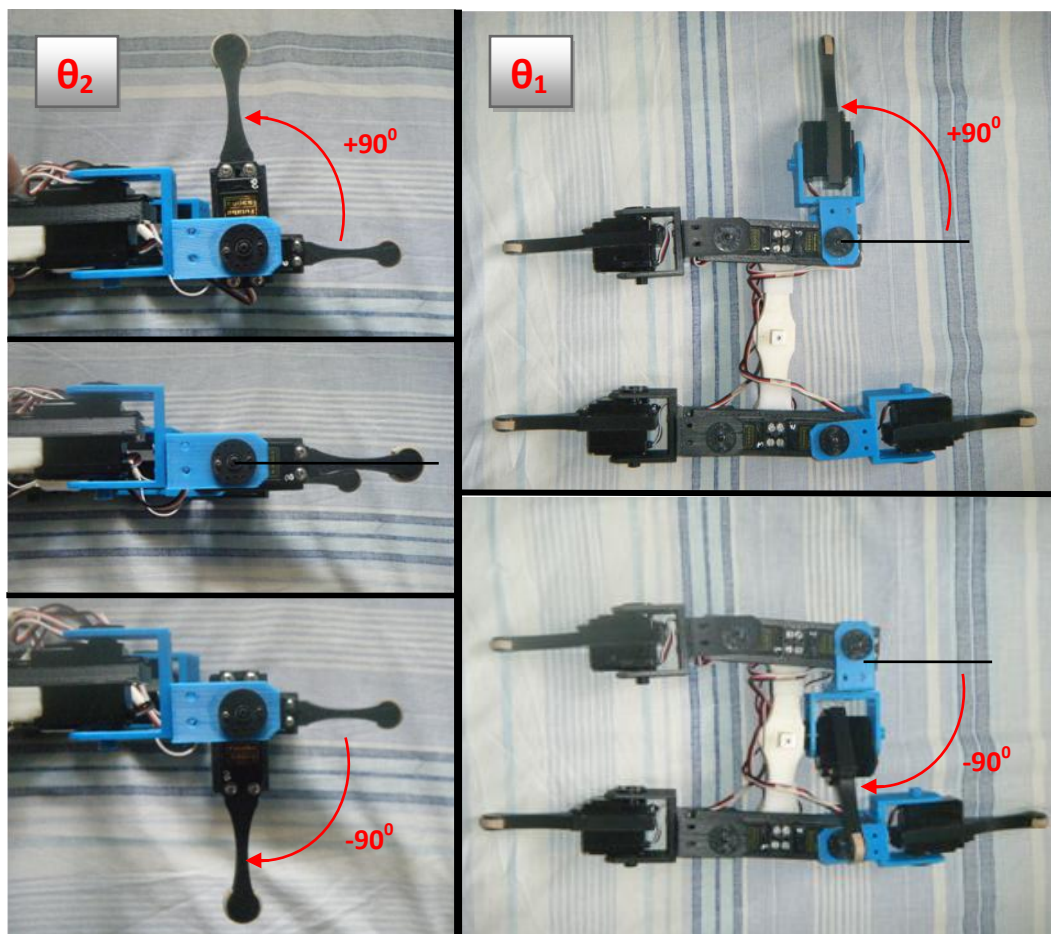


Figura 39 - Ángulos de giro máximos

Estudio de locomoción del AT-AS:

Como se ha mencionado al hablar de los *Cuatro patas – Robots Cuadrúpedos* en el apartado 3.1 de esta memoria, los robots de cuatro patas tienen dos patrones de marcha típicos, los cuales el AT-AS puede seguir al ser también un cuadrúpedo. A continuación se desarrollan con más detalle estos modos de avance:

A. Pata por pata:

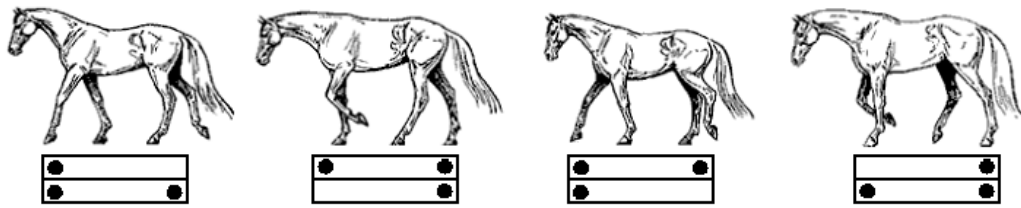


Figura 40 - Caballo caminando avanza en modo ‘pata por pata’

En este tipo de locomoción, hay tres puntos de contacto con la superficie en todo momento (ver ejemplo de un caballo caminando en *Figura 40*), el centro de gravedad (CG) del robot está siempre sobre el triángulo formado por los tres pies apoyados como muestra la siguiente *Figura 41* el AT-AS levantando una de sus patas.

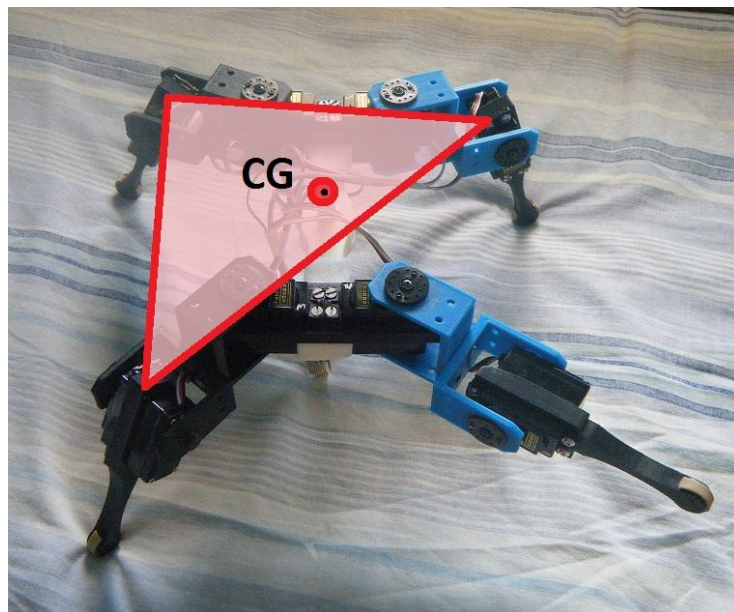


Figura 41 - Centro de Gravedad avanza pata por pata

De esta forma el robot mantiene la estabilidad estática tanto estando quieto como en movimiento. Mover una pierna a la vez hace que el robot avance más lento y gaste más energía, pero lo mantiene estable.

B. Alternando par:

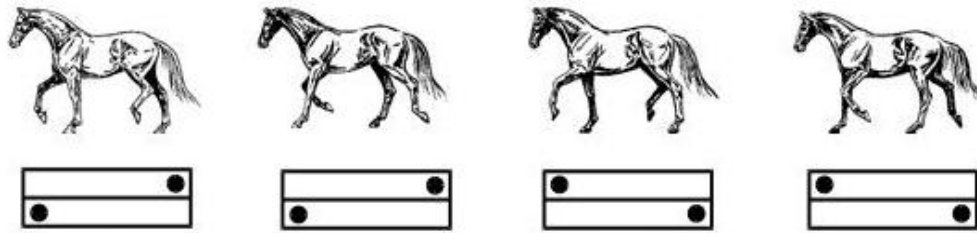


Figura 42 - Caballo trotando avanza en modo ‘alternando par’

En este otro modo, las patas del cuadrúpedo se mueven como si dos bípedos estuvieran conectados entre sí. En cualquier momento el cuadrúpedo tiene dos puntos de contacto con la superficie del suelo. Podemos ver los dos puntos de apoyo marcados en la *Figura 42*, donde se ilustra como ejemplo el avance en modo trote que usan los caballos. Durante este tipo de locomoción, se sitúa el centro de gravedad (CG) sobre una línea equilibrio formada por la unión de los dos puntos de apoyo. La siguiente imagen (*Figura 43*) muestra dicha línea de equilibrio que se forma al levantar dos de las patas del AT-AS.



Figura 43 - Centro de Gravedad avance ‘alternando par’

El modo de avance ‘alternando par’ no es estáticamente estable como el primer modo ‘pata por pata’, presenta pequeños periodos de caída que deben ser cortos y bien controlados. Sin embargo, es mucho más rápido y eficiente en movimiento (estabilidad dinámica).

3.4.DISEÑO DE LAS PIEZAS DE LA ESTRUCTURA IMPRIMIBLE

Una vez elegido el modelo cuadrúpedo, se comenzó a diseñar cada una de las piezas del robot AT-AS. Podemos ver en la *Figura 44*, una composición con todas las piezas necesarias en la estructura de dicho robot.

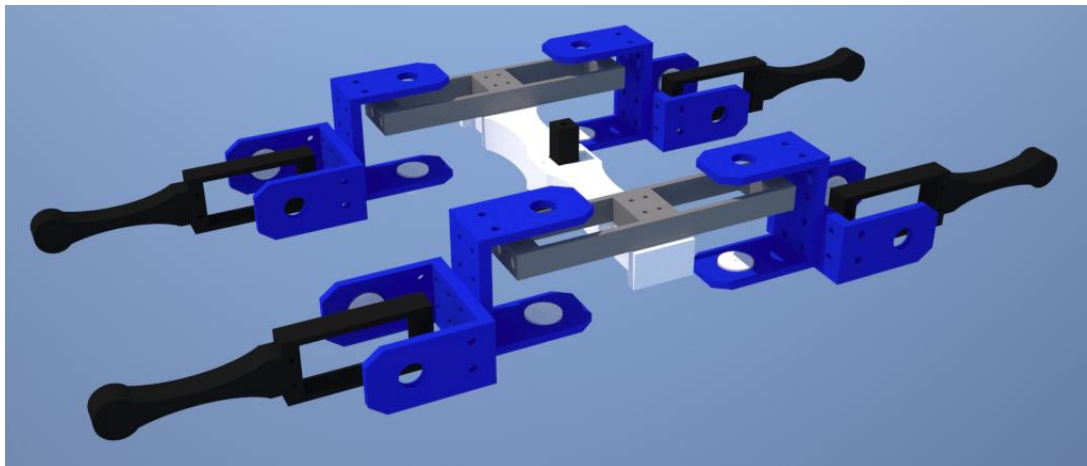


Figura 44 - Piezas imprimibles del AT-AS

Como puede observarse en las imágenes (*Figura 44* y *Figura 45*), el diseño es completamente modular, lo que ofrece claras ventajas a la hora de modificarlo y evolucionarlo. El AT-AS consta de cuatro extremidades, denominadas en el contexto de este proyecto como ‘*piernas*’, y un ‘*esqueleto*’ (que incluye una ‘*columna central*’ que se aprecia de color blanco en dichas imágenes).

Cada *pierna* está formada por cinco piezas: un ‘*pie*’ (en negro) y dos ‘*huesos*’ (azules) con dos pequeños ‘*botones*’ blancos. Las *piernas* se acoplan de dos en dos a la *columna central* gracias a las que se ha optado por llamar ‘*caderas*’ (de color gris).

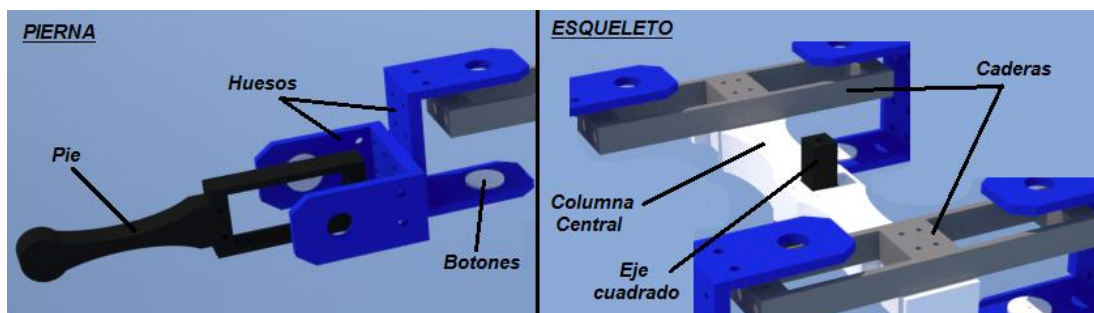


Figura 45 - Esquema de los nombres de las piezas

A continuación, se muestra y explica con detalle cada una de las piezas diseñadas para el AT-AS. Para facilitar mejor su identificación en lo que resta de memoria, las citaremos y nombraremos siempre con los términos anatómicos comúnmente conocidos mencionados en el párrafo anterior e ilustrados en la *Figura 45*.

Diseño de las ‘Piernas’ del AT-AS:

Como se ha mencionado anteriormente, el conjunto llamado ‘pierna’ se compone de las cinco piezas indicadas en la siguiente *Tabla 2*.

Pierna	
Nombre de la pieza	Número de piezas
<i>Pie</i>	1
<i>Huesos</i>	2
<i>Botones</i>	2
TOTAL	5

Tabla 2 - Composición de cada pierna

▪ *Pies:*

Para el diseño de los nombrados ‘pies’ se han utilizado las herramientas que ofrece *OOML* (Librería C++ orientada a objetos para desarrollar diseños mecánicos imprimibles en 3D), gracias a la cual se ha generado el archivo ‘.scad’ (ejecutable por el programa *OpenSCAD*).

En la *Figura 46*, se puede ver el diseño creado con un hueco rectangular del tamaño correcto para adaptar el pie a uno de los servomotores *Futaba S3003*, con sus correspondientes agujeros también a la distancia correspondiente para los tornillos de fijación de los servos.

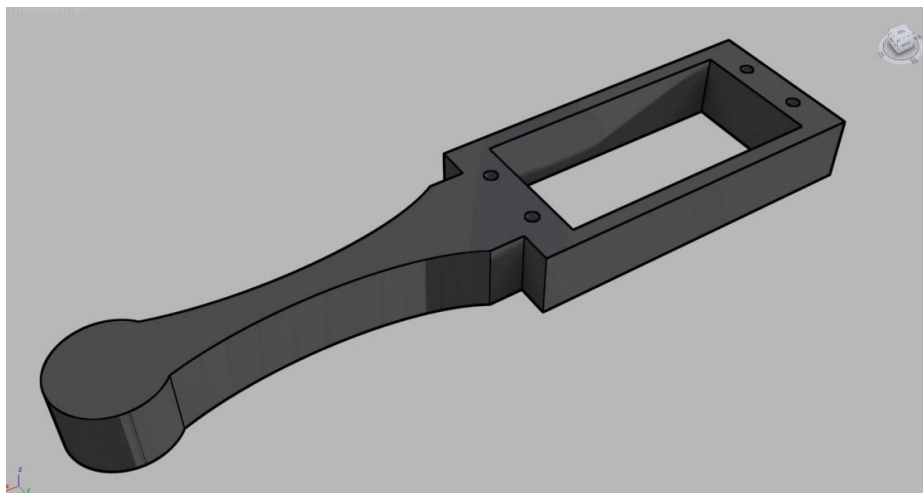


Figura 46 - Diseño del ‘pie’ del cuadrúpedo con hueco para acoplar el servo

▪ **Huesos y botones:**

El diseño de las piezas correspondientes a los ‘*huesos*’ y ‘*botones*’ está tomado del usuario “*bryanandaim*” (7) de la web *Thingiverse*. En una primera fase estos diseños fueron utilizados para facilitar las pruebas de movimiento en las articulaciones impresas en plástico ABS con la intención de realizar con posterioridad un diseño propio. Al comprobar que cumplía sobradamente con su cometido se tomó la decisión de adoptarlas definitivamente y sin modificarlas para el proyecto AT-AS.

En la *Figura 47*, se puede ver una imagen del diseño de uno de los *huesos* y de uno de los *botones*. El hueso tiene el espacio preciso para adaptar en su interior un servomotor *Futaba S3003* y el botón en uno de los agujeros grandes laterales (como se explica más adelante en el apartado 4.1 de esta memoria).

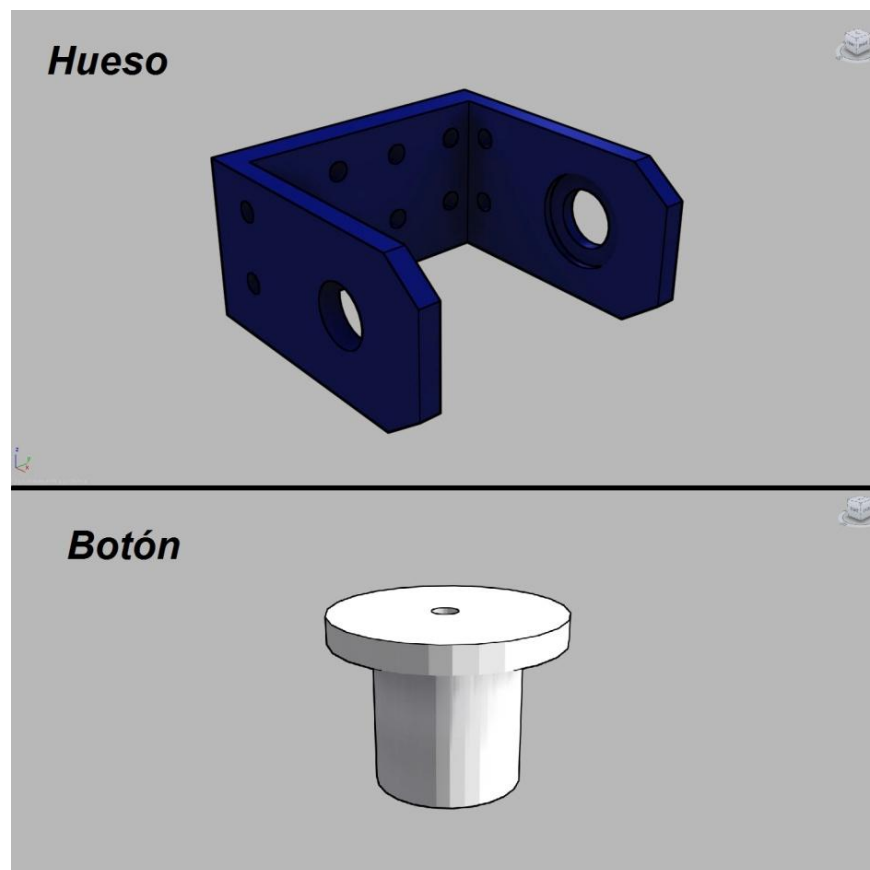


Figura 47 - Diseño de uno de los ‘huesos’ y de un ‘botón’

Diseño del ‘Esqueleto’ del AT-AS:

El conjunto llamado ‘*esqueleto*’ incluye un total de cuatro piezas indicadas en la siguiente *Tabla 3*.

Esqueleto	
Nombre de la pieza	Número de piezas
<i>Caderas</i>	2
<i>Columna central</i>	1
<i>Eje cuadrado central</i>	1
TOTAL	4

Tabla 3 - Composición del esqueleto

▪ *Caderas:*

Las dos *caderas* se han diseñado gracias al programa *OpenSCAD*. Tienen dos huecos rectangulares del tamaño preciso para acoplar servomotores *Futaba S3003*, huecos iguales a los que tienen las piezas de los *pies*. También tienen los agujeros interiores y exteriores correspondientes a los tornillos de fijación de los servos. En la *Figura 48* se observa dicha pieza.

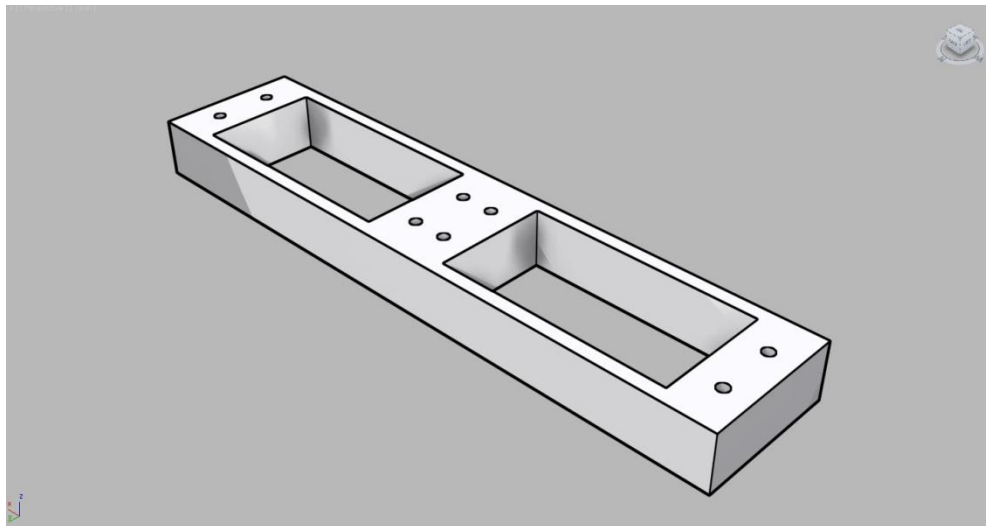


Figura 48 - Diseño de la pieza ‘cadera’ que sujeta las piernas al esqueleto central

- **Columna central:**

La *columna central*, también diseñada en *OpenSCAD*, se ha desarrollado pensando en ser suficientemente robusta para no dar margen a la torsión ocasionada por las fuerzas de empuje de las patas del cuadrúpedo al andar, complicaciones observadas en las primeras pruebas de implementación del AT-AS. A continuación una imagen de la misma (*Figura 49*).

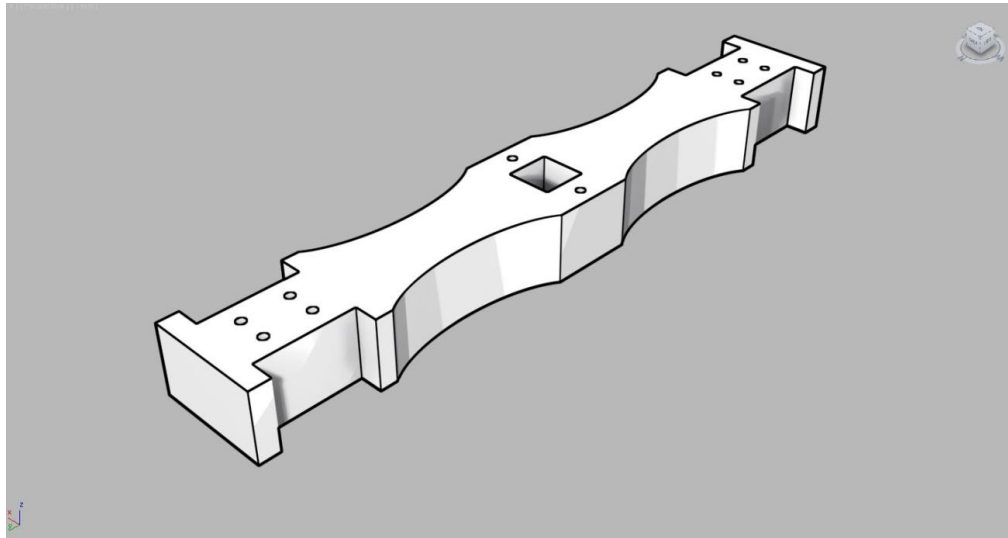


Figura 49 - Diseño de la ‘columna central’

- **Eje cuadrado central:**

La última pieza diseñada (*Figura 50*) para ser impresa en plástico es el *eje cuadrado central*. Una pequeña pieza que por un lado permite soportar la placa Arduino encima del cuadrúpedo, y por otro servir de ayuda a la futura expansión del modelo AT-AS como se contempla en el *Capítulo 5* apartado de “*TRABAJO FUTURO*” de esta memoria.

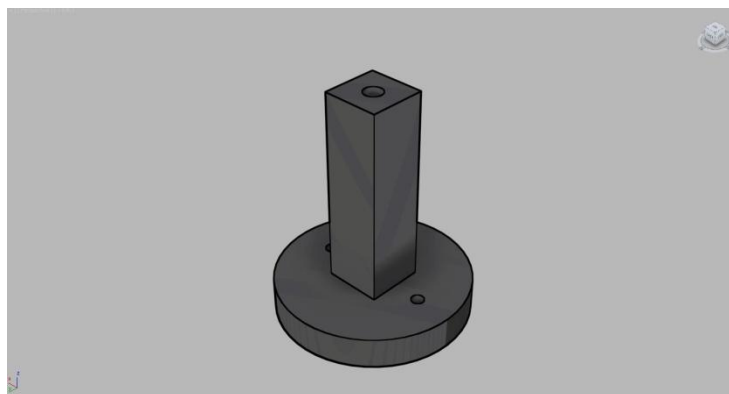


Figura 50 - Diseño del eje central cuadrado

CAPÍTULO 4: FABRICACIÓN E IMPLEMENTACIÓN

4.1. FABRICACIÓN Y MONTAJE

En este apartado se dan detalles y pasos de cómo se han montado todos los elementos que componen el AT-AS, tanto las piezas impresas y servomotores que forman la estructura física como las conexiones eléctricas necesarias para el funcionamiento completo del robot.

Impresión de las piezas en plástico ABS:

Todas las piezas a excepción de la *columna central* fueron impresas en plástico ABS gracias a las impresoras 3D ‘MADRE’ y ‘UC3PO’, ambas modelos *Thing-O-Matic* de la empresa *Makerbot*. La pieza *columna central* al ser demasiado larga para dichas impresoras, se imprimió en ‘GANDALF’ un modelo *RepRap* perteneciente al profesor Alberto Valero.

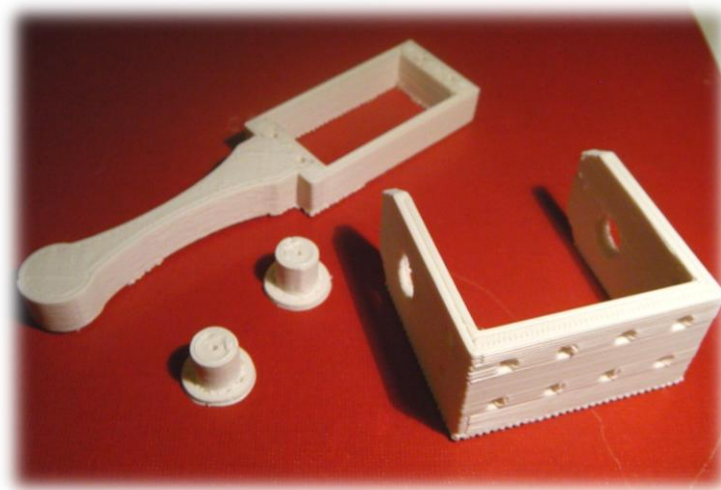


Figura 51 - Piezas impresas en plástico ABS

Montaje de la estructura física:

Una vez impresas todas las piezas del AT-AS, se ha procedido al montaje de la estructura física completa del robot junto con los ocho servomotores, la placa Arduino y la batería LiPo. Para ello se han seguido en orden los pasos que se desarrollan a continuación:

Paso 1: *Montaje del esqueleto central*

En primer lugar, se han montado las cuatro piezas del *esqueleto* del AT-AS, colocándolas tal y como se aprecia en el siguiente esquema (*Figura 52*).

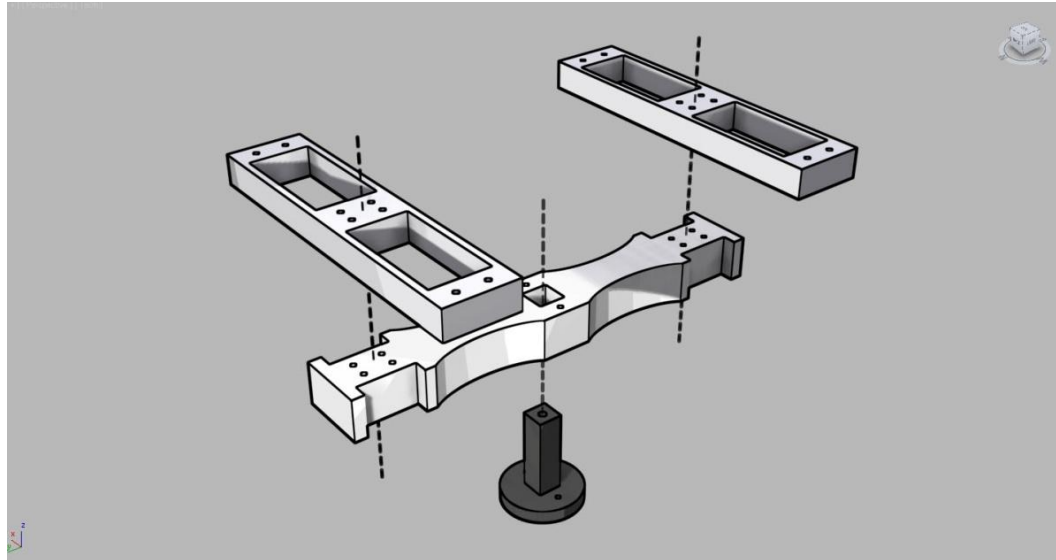


Figura 52 - Diagrama de montaje del esqueleto central

Se ha fijado el *eje cuadrado* (marcado en negro en la *Figura 52*) a la *columna central* gracias a dos tornillos cortos [3mm de diámetro y 1cm de longitud] en los dos agujeros que ambas piezas tienen pensados para este fin.

Para fijar cada *cadera* a la *columna central*, se han utilizado cuatro tornillos largos [3mm de diámetro y 3cm de longitud]. En la imagen que se muestra a continuación (*Figura 53*), se ven marcados los agujeros disponibles para tornillo que existen en la pieza *cadera*.

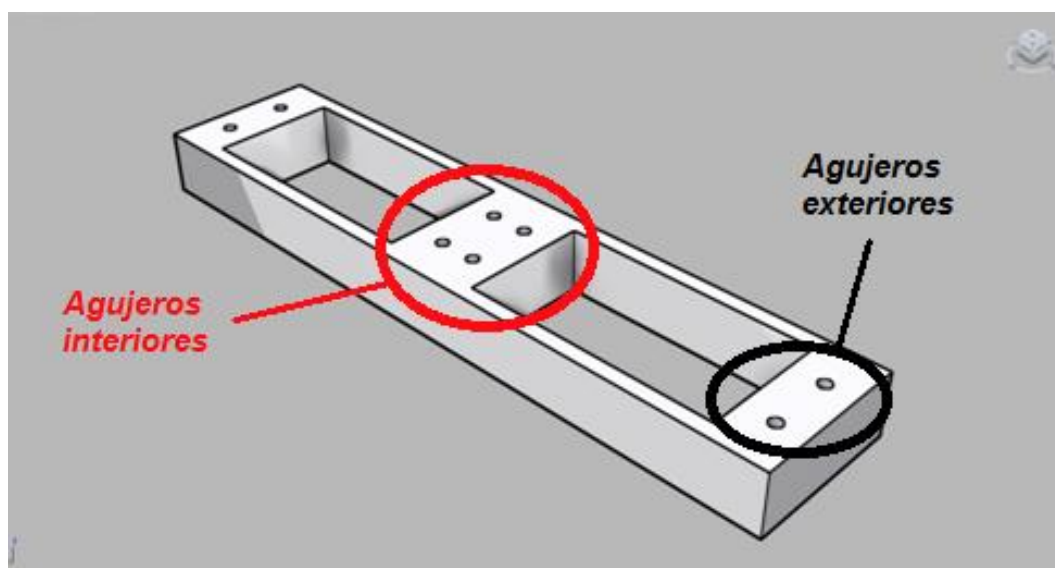


Figura 53 - Detalle de los agujeros para tornillo de las caderas

Los tornillos largos mencionados en el párrafo anterior van a insertarse en los cuatro agujeros señalados como ‘agujeros interiores’, pero justo antes de insertar dichos tornillos en la *cadera*, se han colocado dos de los servos Futaba, ya que estos mismos cuatro tornillos largos sirven para fijar tanto la *cadera* como los servos a la *columna central*. Observando el siguiente diagrama (*Figura 54*) se aprecia la orientación en que van encajados los dos servos y se entiende mejor la función doble de los tornillos largos.

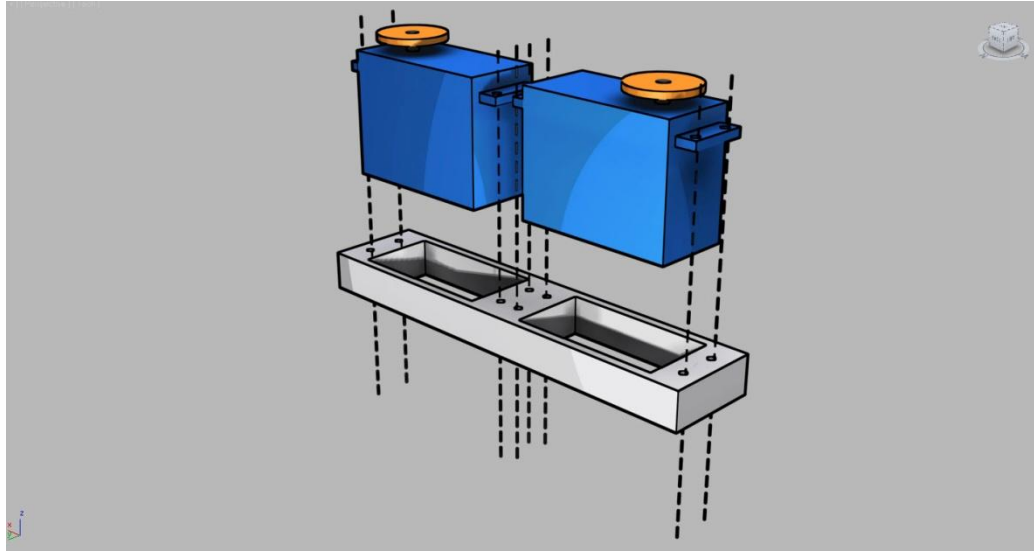


Figura 54 - Diagrama de montaje de los servos en las ‘caderas’ que sujetan las patas

Para finalizar este paso, se ha terminado de fijar los dos servos con otros cuatro tornillos cortos [3mm de diámetro y 1cm de longitud], esta vez insertados en los ‘agujeros exteriores’ de la *cadera* (mirar de nuevo la *Figura 53*).

Paso 2: Montaje de los pies

En el segundo paso, se ha acoplado un servo en la pieza que hará de *pie*, en el hueco correspondiente como puede verse en el siguiente esquema (Figura 55).

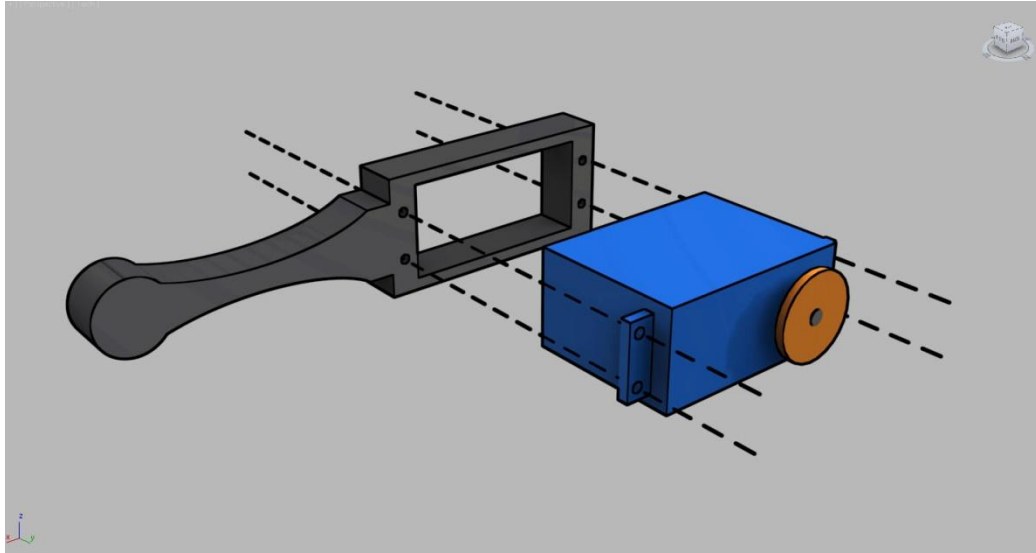


Figura 55 - Diagrama de montaje del servo dentro del 'pie'

Para anclar dicho servo y que no se salga durante el movimiento del AT-AS, se han insertado tornillos cortos fijadores [3mm de diámetro y 1cm de longitud] en los cuatro agujeros del *pie*.

Paso 3: Montaje de los huesos

Los dos *huesos* de la pata se han unido tal y como indica el siguiente diagrama de la Figura 56, usando 4 tornillos fijadores. Los pequeños *botones* son insertados en los huecos grandes de los huesos aunque sin estar fijados ni pegados a los mismos.

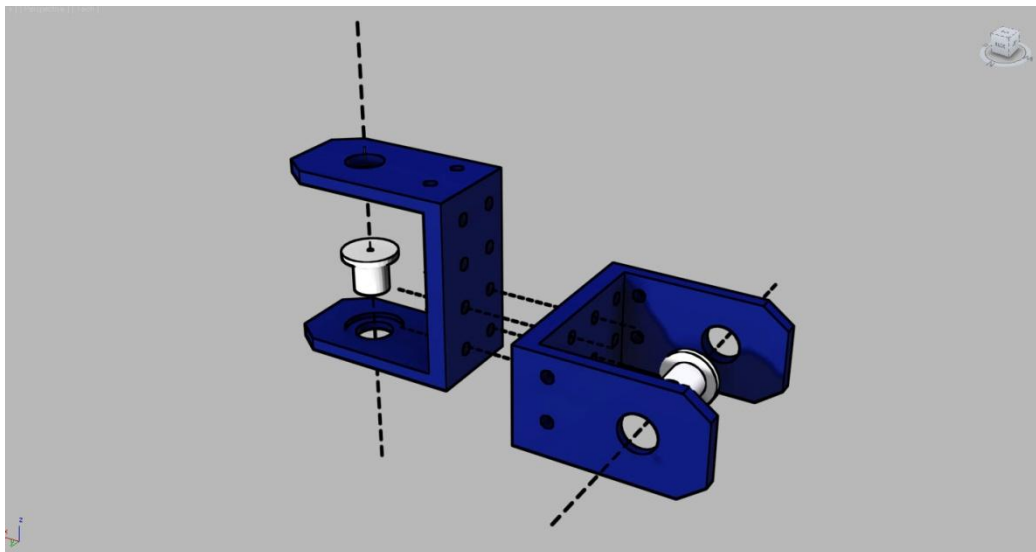


Figura 56 - Diagrama de montaje de los 'huesos' entre sí con la posición de los botones

Paso 4: Montaje de la pierna.

Para el montaje total de la *pierna*, se han sujetado los servomotores dentro de cada articulación. En primer lugar se han **fijado los discos de los servos a cada hueso** gracias a dos pequeños tornillos fijadores [1mm de diámetro y 6mm de longitud], para ello se ha practicado agujeros de 1mm de diámetro gracias a un mini-taladro. El disco queda fijado como se observa en la *Figura 57*.

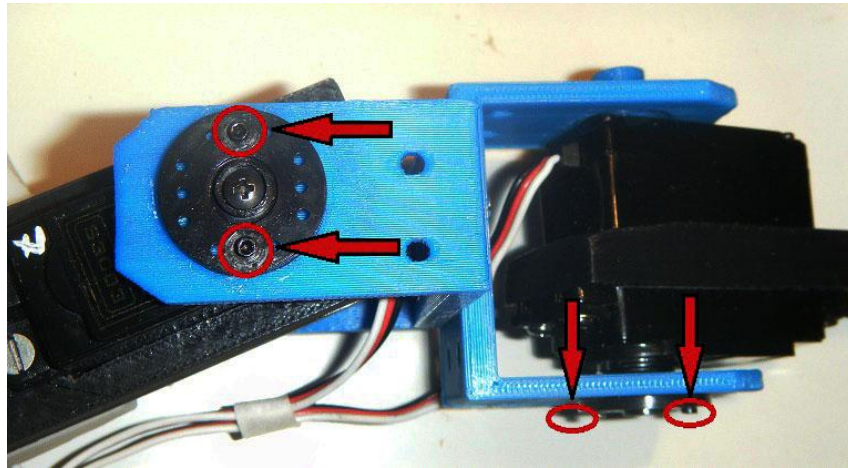


Figura 57 - Fijación de los discos de los servomotores

En segundo lugar, **se han pegado los botones a la parte trasera de los servomotores**, aplicando pegamento en la zona marcada en rojo en la siguiente imagen (*Figura 58*).

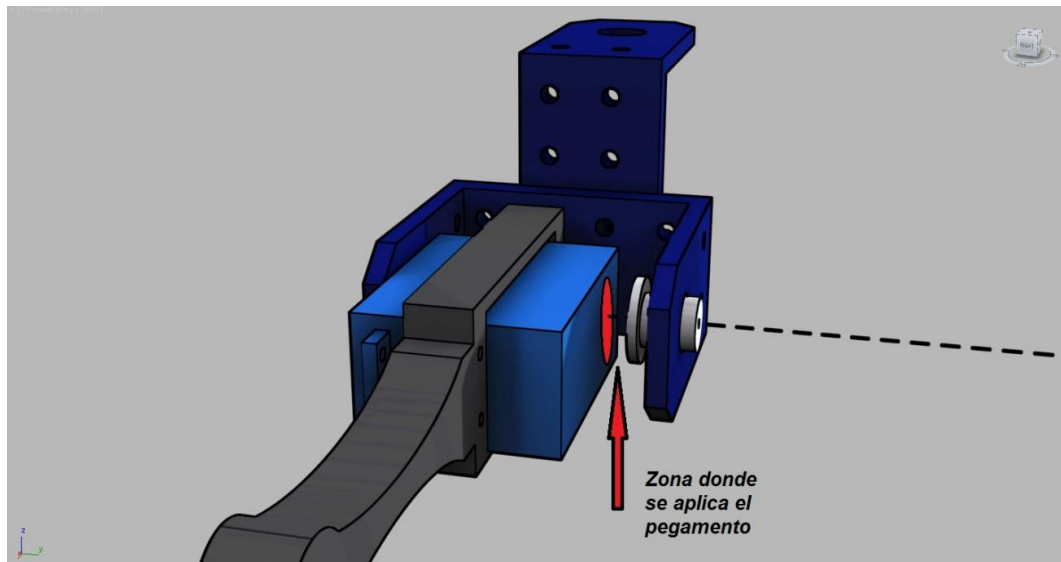


Figura 58 - Montaje de la articulación del pie. Detalle de cómo pegar el botón

El botón debe pegarse a una altura alineada con el eje de giro del disco del servo, tal y como se muestra en la siguiente *Figura 59*.

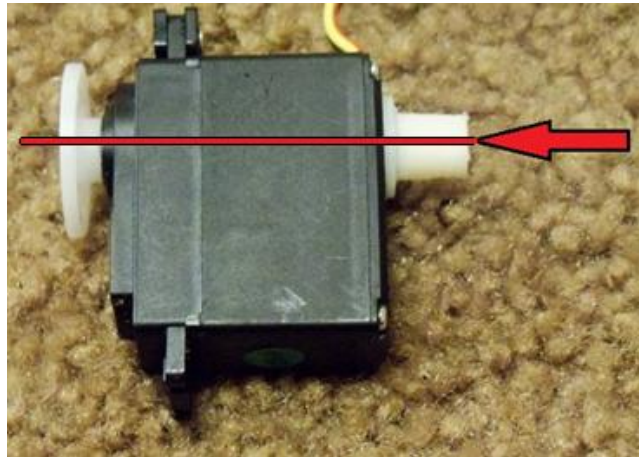


Figura 59 - Detalle de cómo debe quedar el botón

Una vez fijados los dos servomotores (el del *pie* y el de la *cadera*) dentro de las articulaciones de los *huesos*, se ha obtenido una pierna completamente montada como se ve en las siguientes imágenes (*Figura 60*).

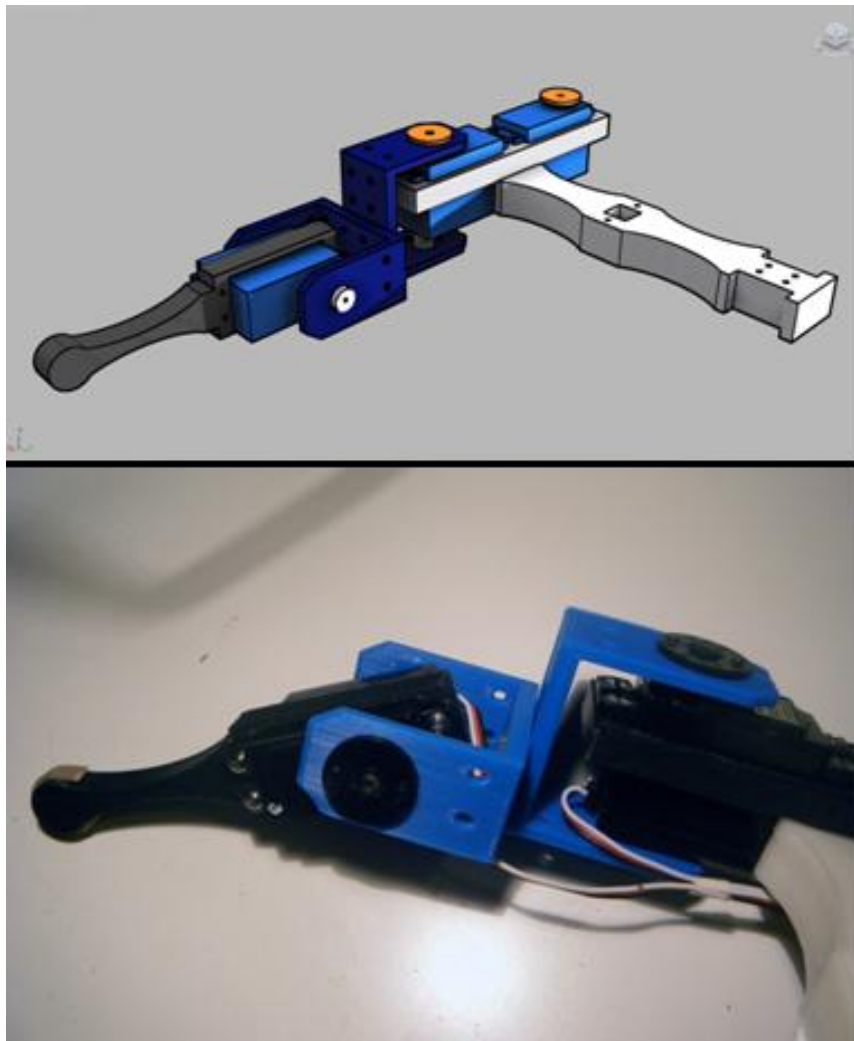


Figura 60 - Pierna montada al completo

Paso 5: *Montaje completo del resto de las piezas impresas*

Para completar el montaje de todas las piezas impresas y los servos, se ha repetido el *Paso 2*, el *Paso 3* y el *Paso 4* con cada una de las patas del cuadrúpedo, de tal forma que la estructura física del AT-AS queda como puede verse en la siguiente imagen (*Figura 61*).

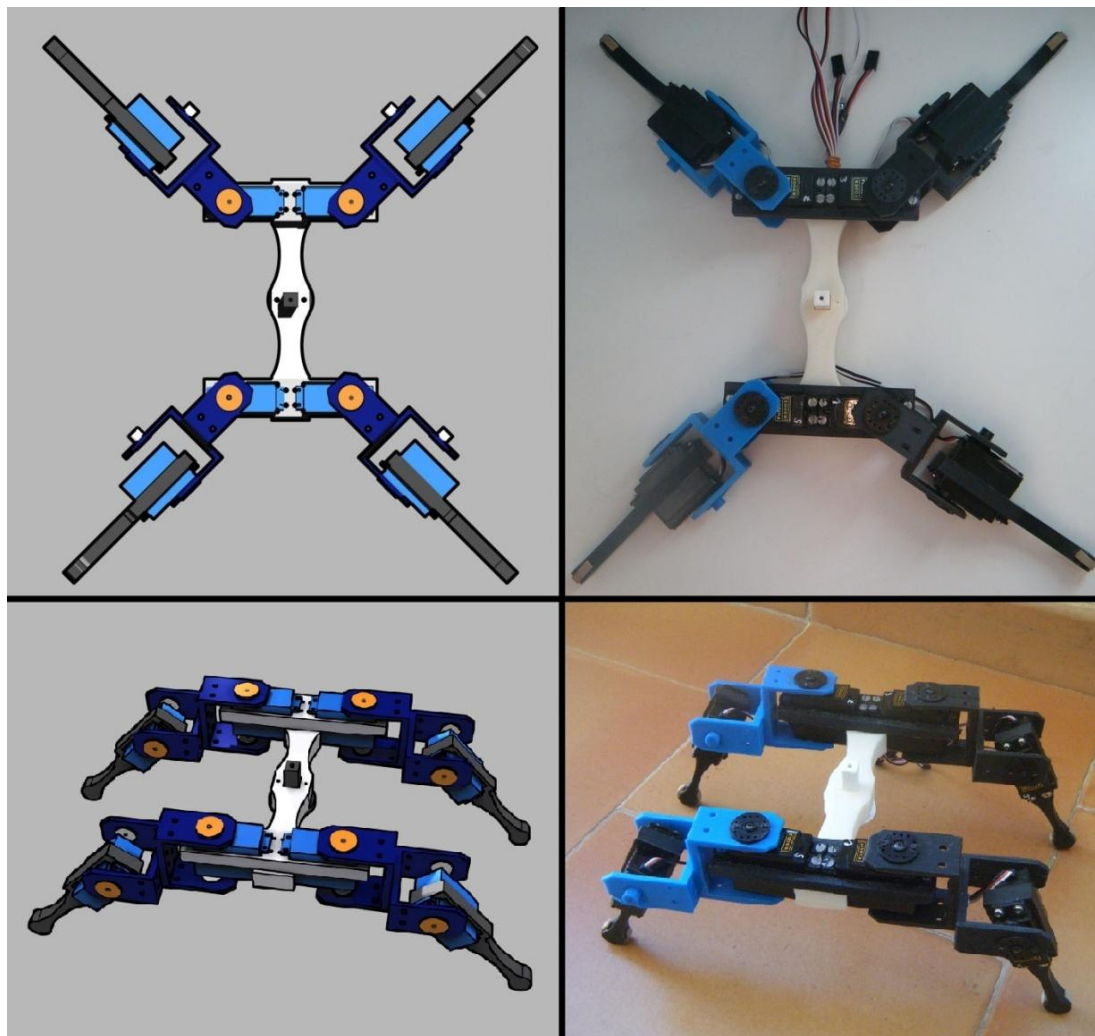


Figura 61 - Vista del montaje completo de todas las piezas impresas junto a los servos

Paso 6: *Montaje final*

Como último paso, se han colocado los componentes electrónicos que faltaban por instalar en la estructura: la placa Arduino *Mega 2560* y la batería LiPo de 7,4V (2S).

La **placa Arduino Mega 2560** ha sido fijada encima del AT-AS por medio de un tornillo largo [3mm de diámetro y 3cm de longitud] insertado en el *eje cuadrado central*. Dicho tornillo junto con dos tuercas y dos arandelas de goma para no dañar la placa, permite elevar y bajar la placa Arduino hasta un centímetro si se tuviese que

modificar la altura donde va instalada. En la *Figura 62* se muestra el orificio de la pieza *eje cuadrado central* donde se coloca dicho tornillo largo así como unas fotos de cómo ha quedado instalada la placa.

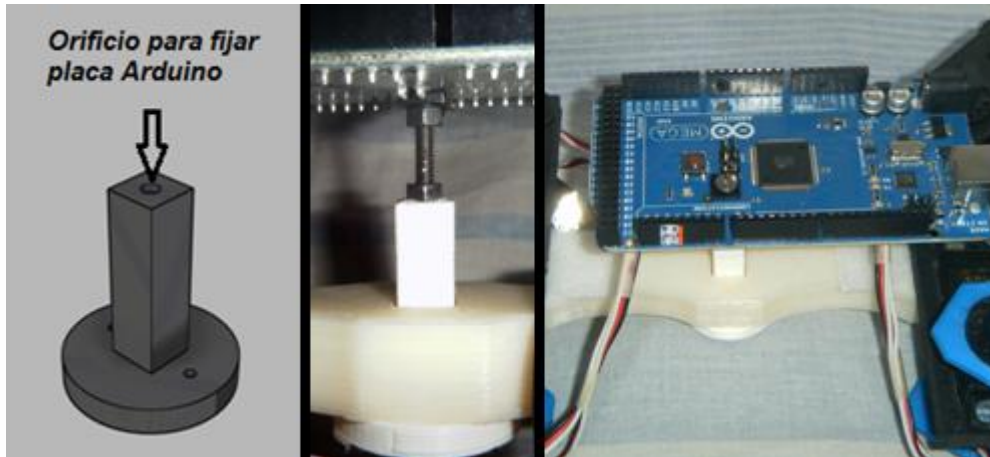


Figura 62 - Detalle del orificio para tornillo del eje cuadrado central

Para colocar la **batería LiPo** bajo el AT-AS, se han utilizado tiras de velcro. Las tiras de velcro permiten separar la batería LiPo de la estructura del cuadrúpedo cuando se vaya a cargar/descargar o cuando el AT-AS vaya a guardarse (es recomendable en ambos casos separar la LiPo para guardarla en lugar seguro, a salvo de posibles golpes o cambios bruscos de temperatura).

Tres de las tiras de velcro (del tipo con superficie suave) abrazan la batería como se puede ver señalado en la siguiente imagen (*Figura 63*). Además, se han fabricado otras dos tiras largas (en este caso del tipo con superficie rugosa, señaladas en la *Figura 64*) de 17cm de longitud, con el objetivo de sujetar la batería al esqueleto del robot.

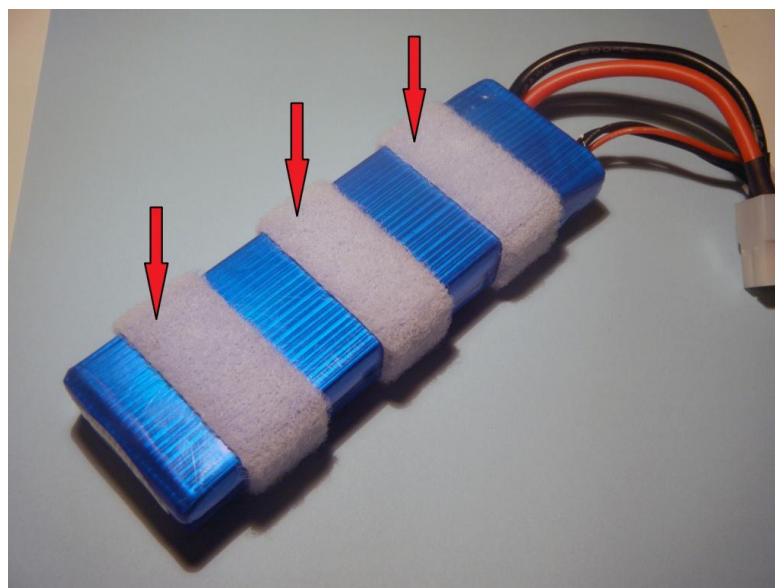


Figura 63 - Tres tiras de velcro abrazando la batería Lipo

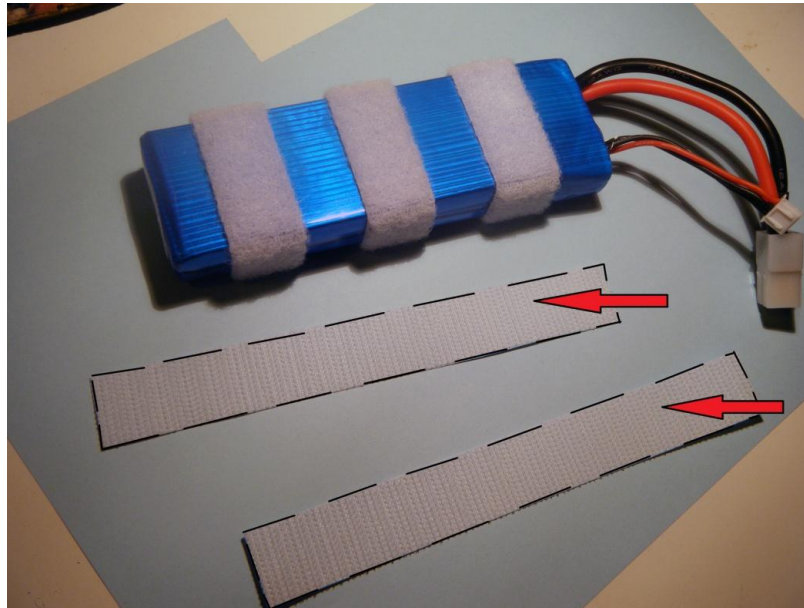


Figura 64 - Dos tiras de velcro largas

Además, otros tres pequeños trozos de velcro se han adaptado al esqueleto principal del robot. Como se ve en la *Figura 65*, un pequeño cuadrado (del tipo con superficie rugosa) se ha pegado en la parte baja donde se apoya posteriormente la batería LiPo.

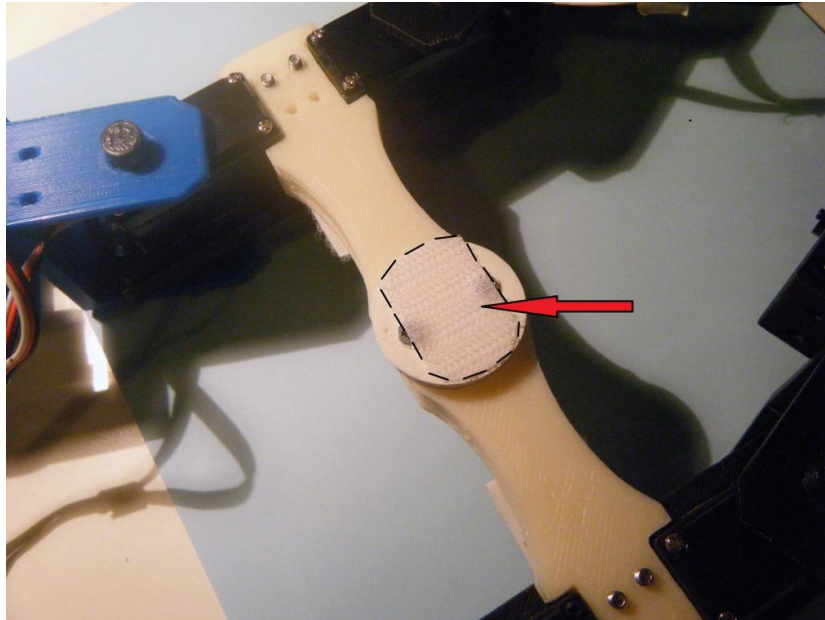


Figura 65 - Detalle de velcro en la parte baja

Y por último, dos pequeños cuadrados más de velcro (del tipo con superficie suave), se han pegado a la parte alta del esqueleto central en las dos zonas señaladas de la *Figura 66*.

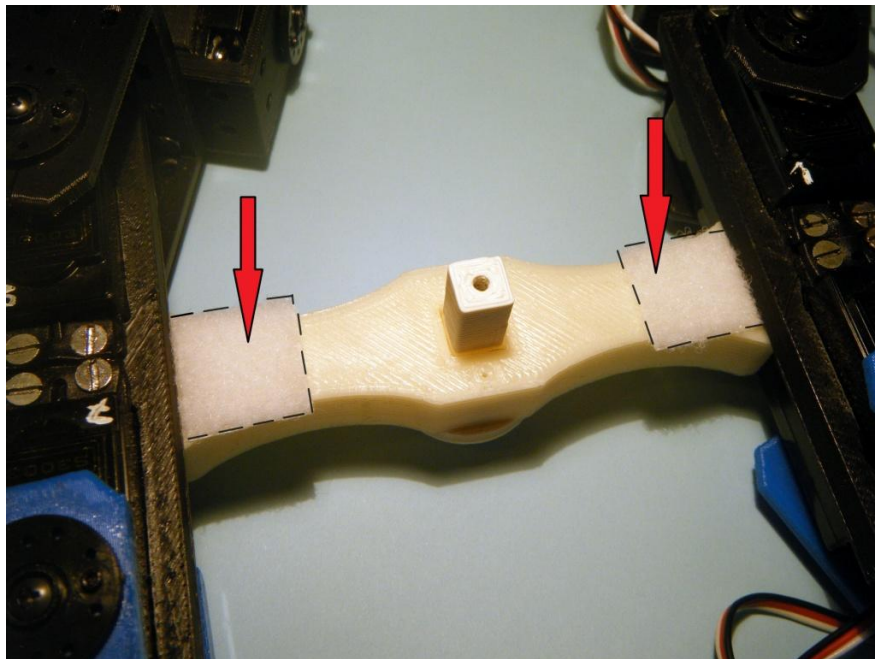


Figura 66 - Detalle de velcro en la parte alta

La batería LiPo queda instalada de forma segura y fija con el velcro como puede verse en la siguiente foto (*Figura 67*).

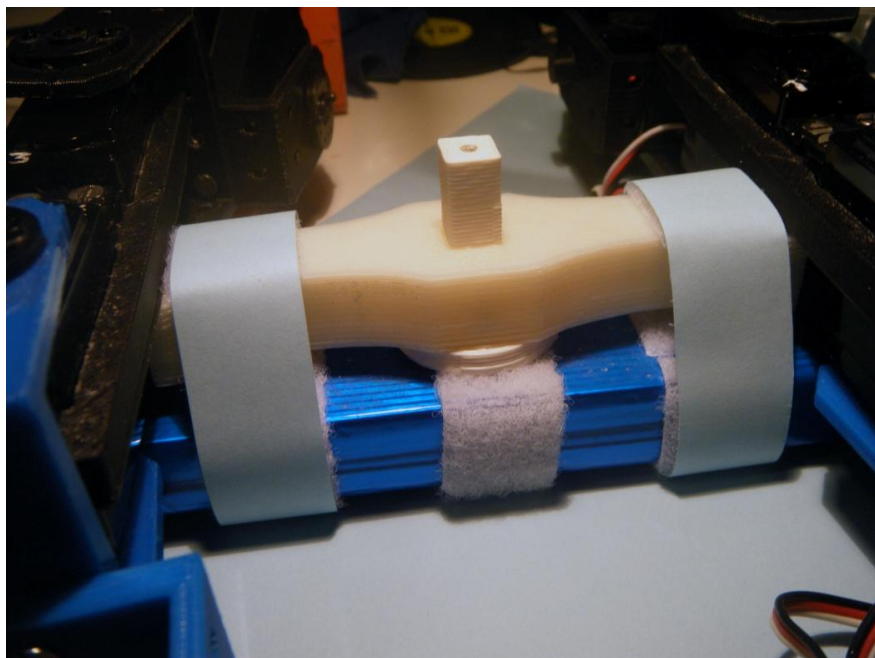


Figura 67 - Montaje final de la batería LiPo

Conexiones eléctricas:

A continuación se describen las conexiones eléctricas necesarias para el correcto funcionamiento de todos los elementos del proyecto.

▪ *Conexiones de la alimentación:*

Para alimentar todos los elementos eléctricos del AT-AS, se ha seguido el esquema de la *Figura 68*. Como se puede observar, la placa *Arduino Mega 2560* va directamente conectada a la batería LiPo, ya que soporta los 7,4V. A su vez, la placa Arduino alimenta los módulos inalámbricos que se conectan gracias a las salidas de 5V y GND disponibles en la placa.

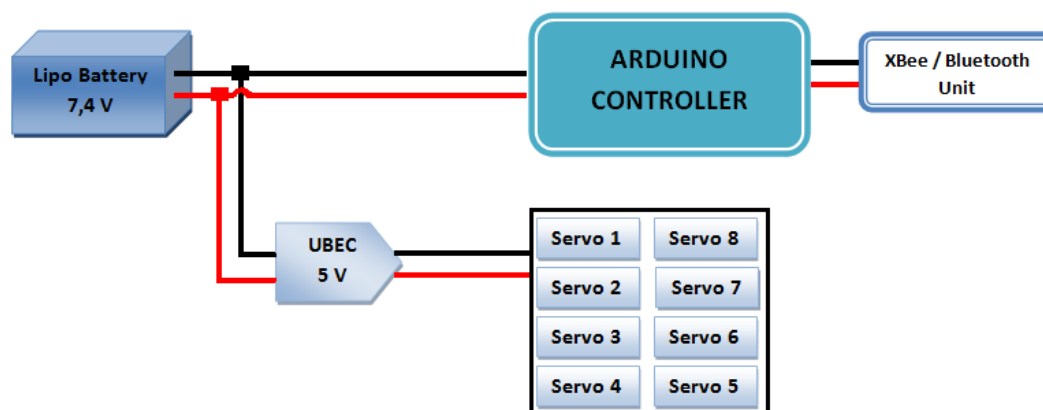


Figura 68 - Esquema de conexión de la alimentación del AT-AS

Los servomotores en cambio, se estropearían si fuesen conectados directamente a la batería LiPo ya que no soportan los 7,4V. Sin embargo, tampoco se podrían conectar directamente a la salida de 5V de la placa Arduino porque esta salida no puede ofrecer más de 50mA y con la intensidad que demandan 8 servomotores podría quemarse. Por ello se han conectado todos los servos a la salida del UBEC que les proporciona 5V (o 6V seleccionable mediante una pestaña del UBEC), tensión a la que funcionan correctamente.

En la *Figura 69* se muestra una imagen de los cables utilizados y los conectores soldados a dichos cables. En la misma imagen se observa cómo se conecta la placa Arduino y también la batería LiPo gracias a los adaptadores correspondientes. A su vez, para adaptar el conector de salida del UBEC a los 8 servomotores se ha fabricado un pequeño circuito que se explica en el siguiente apartado.

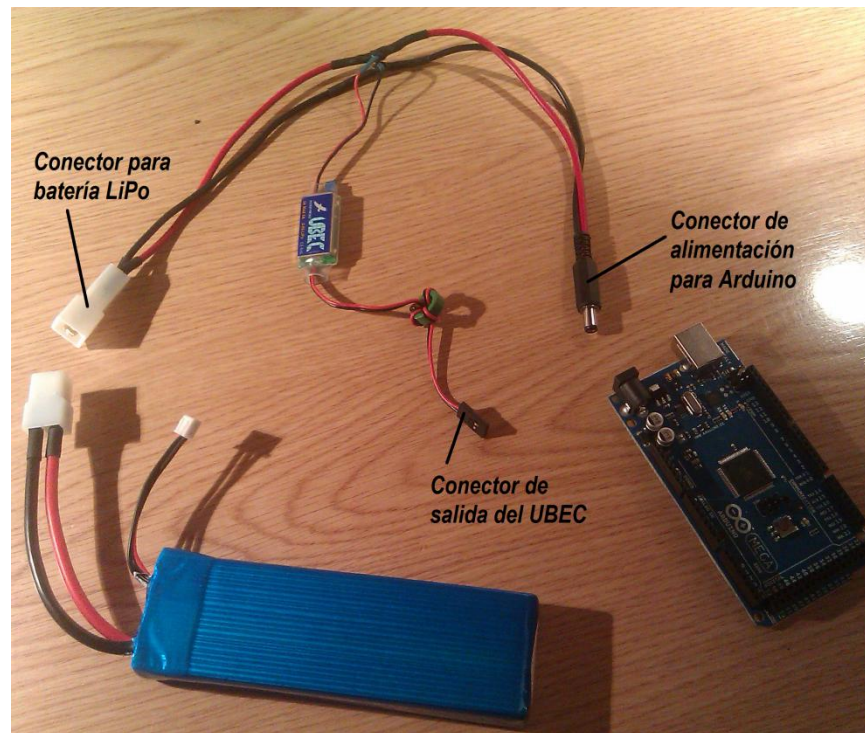


Figura 69 - Foto de los cables y conectores eléctricos

- **Conexiones de los cables de control de los servomotores:**

Tanto para la conexión de la alimentación como para la conexión de los cables de control de los servomotores se ha fabricado el pequeño circuito que se muestra en la Figura 70.

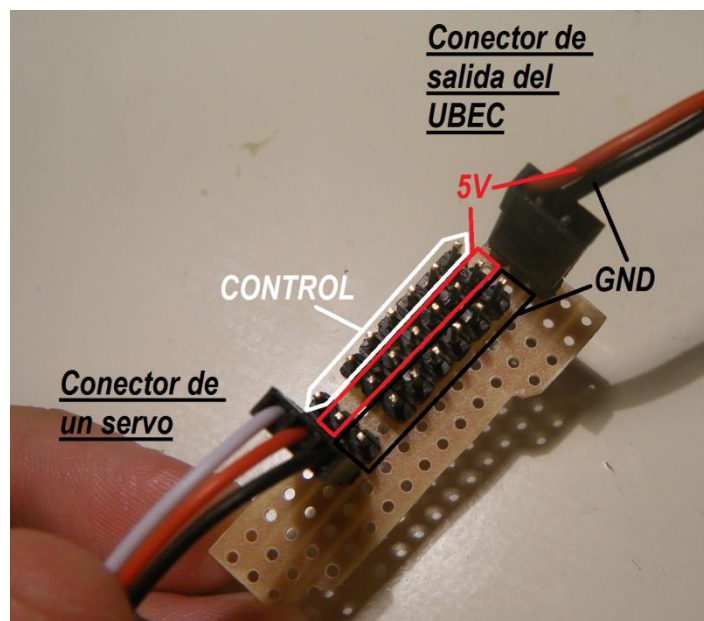


Figura 70 - Circuito alimentación y control servos

Todos los pines marcados con un recuadro rojo en la figura anterior (5V), están soldados entre sí por debajo del circuito, al igual que todos los pines marcados con un recuadro negro (GND). Los pines marcados en blanco corresponden a los cables de control de los servos. Dichos pines no están conectados entre sí y son el doble de largos para así poder ser conectados directamente a las salidas digitales de la Arduino Mega 2560. En la *Figura 71* se pueden ver los pines largos salientes por debajo del *mini* circuito, así como los pines cortos soldados por filas.

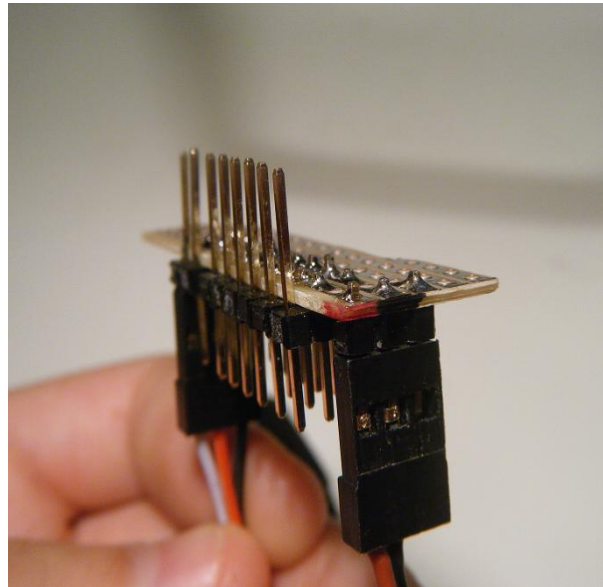


Figura 71 - Vista de los pines largos del mini circuito

En la siguiente imagen (*Figura 72*), se ha conectado el *mini* circuito a las salidas digitales de la placa Arduino, abarcando todos los huecos para pines entre la salida digital número 2 y la salida número 9.

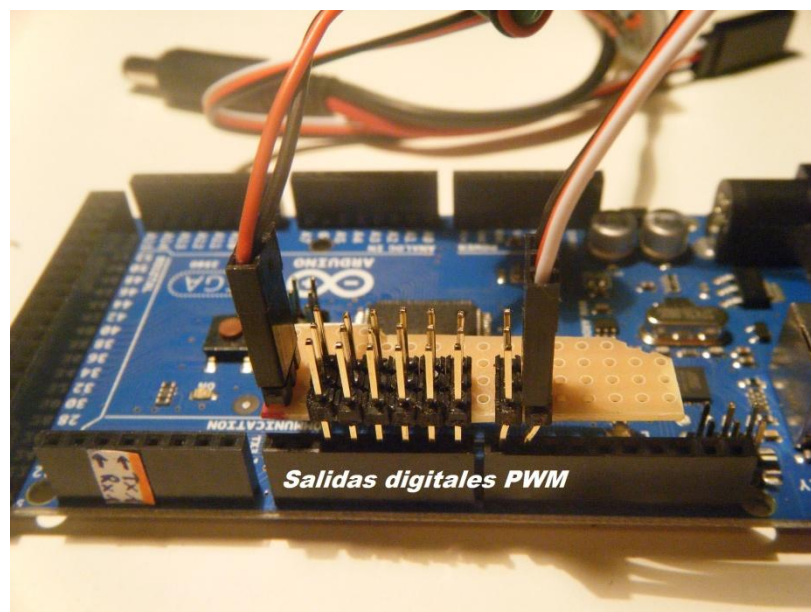


Figura 72 - Conexión mini circuito con salidas digitales de Arduino

Para finalizar, se han conectado todos los cables de los 8 servomotores al mini circuito. Para ello, se han numerado los servos del AT-AS como se señala en la siguiente *Figura 73*.

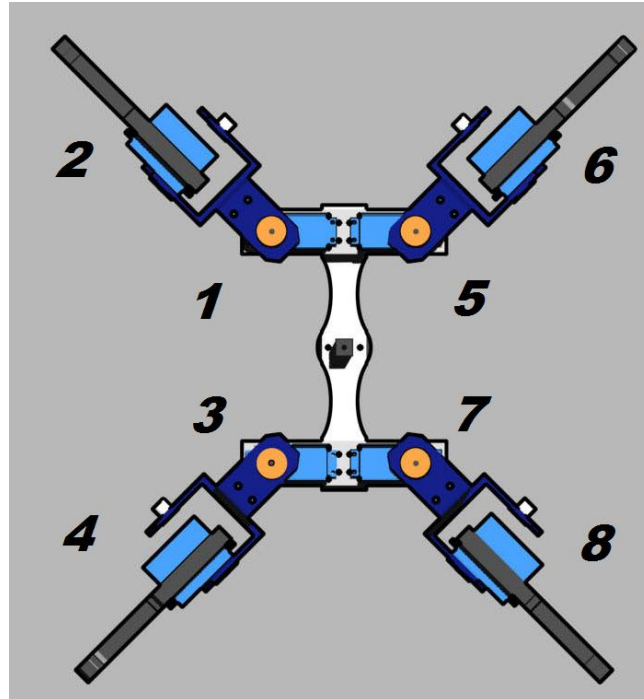


Figura 73 - Numeración de los servos

Y tras numerar los servos, se han conectado los cables de los servos a los pines del *mini* circuito de forma que se cumplan las conexiones de control indicadas en la siguiente *Tabla 4*. Los números de servo correspondientes a cada pin digital se han decidido por la proximidad física de cada servo a dicho pin.

NÚMERO SERVO	PIN DIGITAL ARDUINO
1	9
2	8
3	6
4	7
5	2
6	3
7	4
8	5

Tabla 4 - Conexión de Servos con salidas digitales Arduino

▪ *Conexión del Módulo Bluetooth:*

Para utilizar el módulo Bluetooth, se siguieron las siguientes indicaciones: después de subir el *sketch* (programa desarrollado) a la placa Arduino *Mega 2560* por medio del puerto serie (cable USB estándar), fue desconectada la placa y se conectaron los pines del módulo bluetooth cruzando *Rx* y *Tx*, de forma que se cumpliesen los siguientes puntos:

1. RXD del módulo bluetooth conectado a TX1 de la placa Arduino.
2. TXD del módulo bluetooth conectado a RX1 de la placa Arduino.
3. VCC del módulo bluetooth conectado a un pin de 5V de salida de la placa Arduino.
4. GND del módulo bluetooth conectado a GND de la placa Arduino.

En la siguiente imagen (*Figura 74*) se muestra un esquema de conexionado del módulo *Bluetooth*.

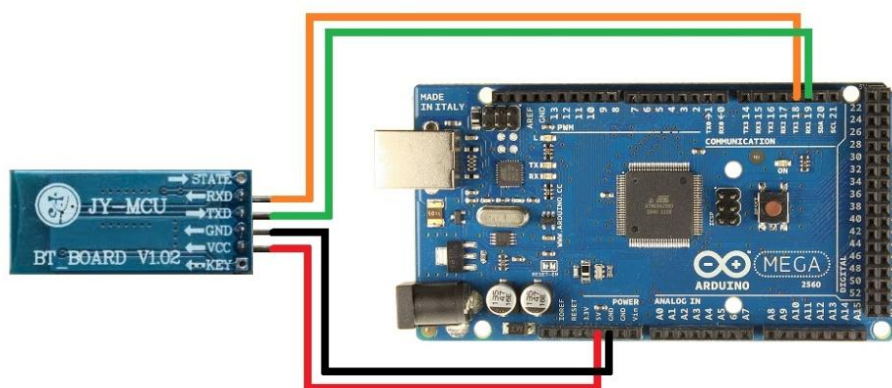


Figura 74 - Conexión del módulo Bluetooth

▪ *Conexión de los Módulos Xbee:*

Cabe recordar que las conexiones que se explican a continuación son alternativas al uso del Bluetooth, pues no se han utilizado ambos métodos de comunicación inalámbrica a la vez.

La instalación de éstos módulos es sencilla, basta con conectar el primer módulo al adaptador USB llamado *Xbee Explorer* y el segundo módulo *Xbee* al adaptador *Xbee Shield para Arduino*. Dicho adaptador *Xbee Shield* se instala en la placa *Arduino Mega 2560* en la única posición que acepta. En la siguiente imagen (*Figura 75*) se puede ver cómo queda instalado.

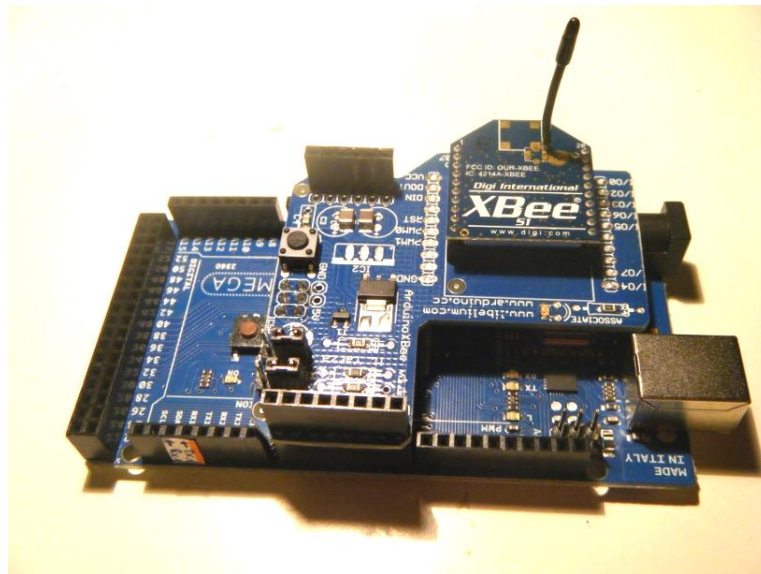


Figura 75 - Instalación Xbee Shield en Arduino Mega 2560

4.2. ENVÍO DE DATOS Y CONTROL REMOTO

Una vez montada la estructura física del AT-AS junto con los elementos electrónicos y conexiones correspondientes, se pudo proceder configurar todos los dispositivos para el control remoto e inalámbrico del cuadrúpedo.

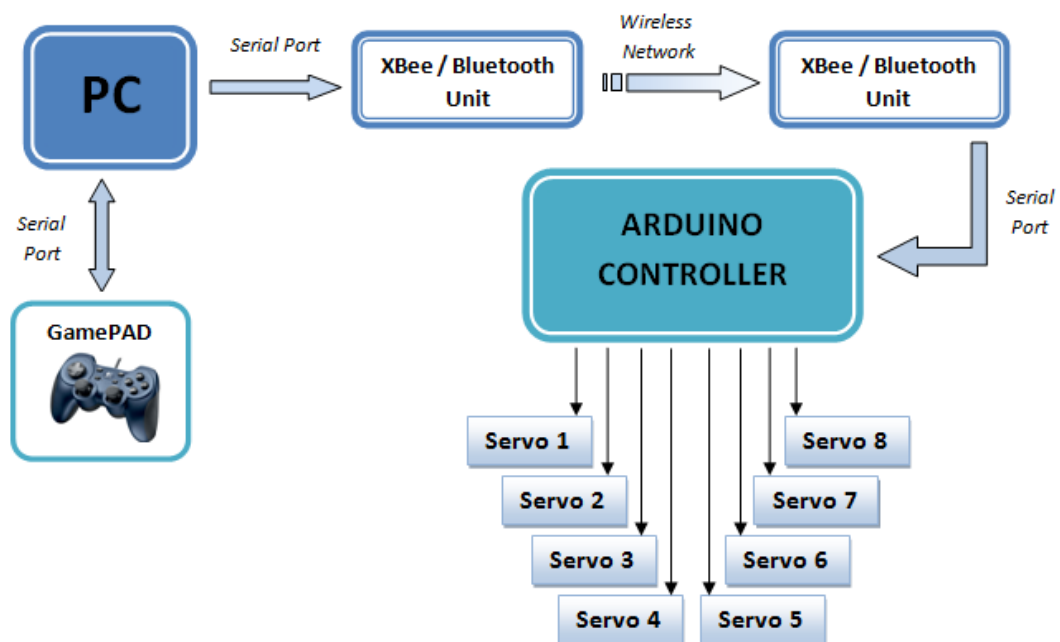


Figura 76 - Esquema de comunicaciones entre elementos del proyecto

En la imagen anterior (*Figura 76*), se muestra un esquema de comunicaciones entre los elementos del proyecto. El *Gamepad* está conectado al PC por medio de un puerto USB. El PC a su vez, puede enviar información a la placa Arduino gracias a los módulos inalámbricos. La placa Arduino recibe información de dichos módulos y envía señales de control a los servomotores.

En los siguientes puntos de este apartado se explica con más detalle las funciones de cada dispositivo y programa desarrollado.

Programa de control en C++:

Para programar y compilar el programa en lenguaje C++ se ha utilizado el entorno *Qt Creator*. En la siguiente imagen (*Figura 77*), se puede ver el esquema general de las principales funciones que realiza dicho programa de control C++.

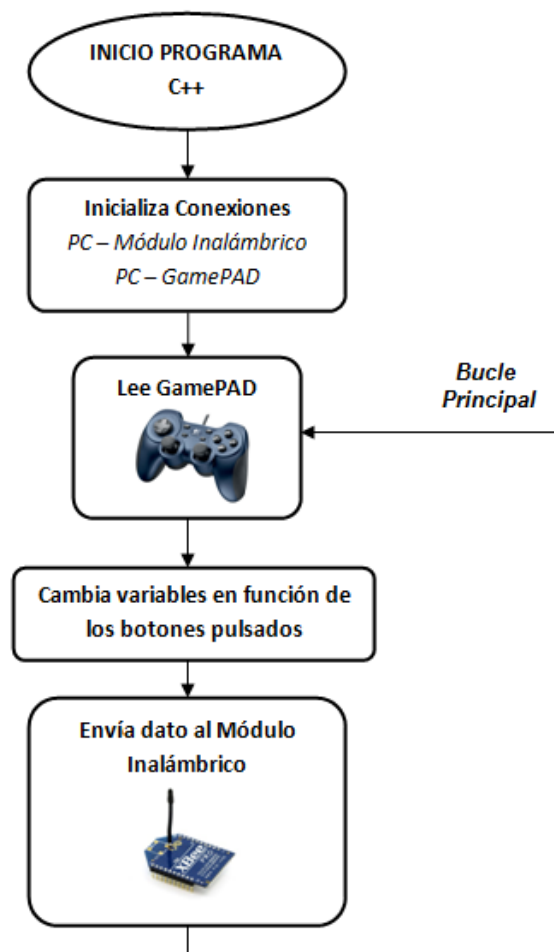


Figura 77 - Esquema general del programa C++

Este programa C++ ejecuta en orden las siguientes acciones:

1. **Comprobación de las conexiones por puerto serie:** la conexión *PC-Gamepad* y la conexión *PC-módulo inalámbrico*.
2. **Lectura de los sensores del mando *Gamepad*** gracias a la clase “*Joystick*” adaptada de un programa realizado por Alberto Valero y Julián Marín para el control remoto de robots. Esta clase calibra las señales recibidas desde el *Gamepad* y ayuda a obtener valores enteros de la posición de los *joysticks* y de los botones del mando.
3. **Procesar la información leída para detectar el movimiento que el usuario quiere realizar con el robot** calculando según los botones pulsados del *Gamepad* los ángulos correspondientes que deberán tomar los servomotores en cada momento para ejecutar ese movimiento.
4. **Modificar la variable “*dato*”** escribiendo en ella dichos ángulos en forma de cadena de caracteres. Se genera el *string* con la siguiente configuración:
 "XXXX.BXXX.CXXX.DXXX.EXXX.FXXX.GXXX.HXXX."
 Incluyendo ocho letras “A, B, C...” separadas entre sí por puntos y seguidas por los valores “XXX” que corresponden a los ángulos calculados para cada servo.
5. **Enviar dicho “*dato*” al módulo inalámbrico** correspondiente por medio del *puerto serie* correspondiente.

Programa de control Arduino:

El programa de Arduino, se ha programado y compilado gracias en la aplicación *Arduino 0019* y posteriormente se ha subido a la placa *Arduino Mega 2560*.

Este programa dentro de la placa Arduino ejecuta en orden las siguientes acciones:

1. **Abrir el *puerto serie*** correspondiente a donde está conectado el módulo inalámbrico con la placa controladora. Abre dicho el *puerto serie* a una velocidad de 9600 baudios.
2. **Crear 8 objetos *Servo* y enlazar dichos objetos con los pines de la *Arduino Mega 2560*** en los que están conectados los cables de control de los servomotores del AT-AS como indicábamos en la *Tabla 4*. Esta acción es posible gracias a la librería de *Servo.h* de Arduino.
3. **Esperar a recibir el “*dato*” procedente del *puerto serie*.**

4. **Decodificar dicho “dato” y enviar las señales de control correspondientes a los servos.** Una función del programa decodifica el *string* recibido el cual sigue teniendo la siguiente configuración:

“AXXX.BXXX.CXXX.DXXX.EXXX.FXXX.GXXX.HXXX.”

Para decodificar el dato recibido, la función separa los caracteres entre punto y punto, y envía el valor del ángulo “XXX” al servo que identifique la letra mayúscula que tiene delante.

A continuación (Figura 78) se puede observar un esquema de las principales acciones que ejecuta el programa Arduino:

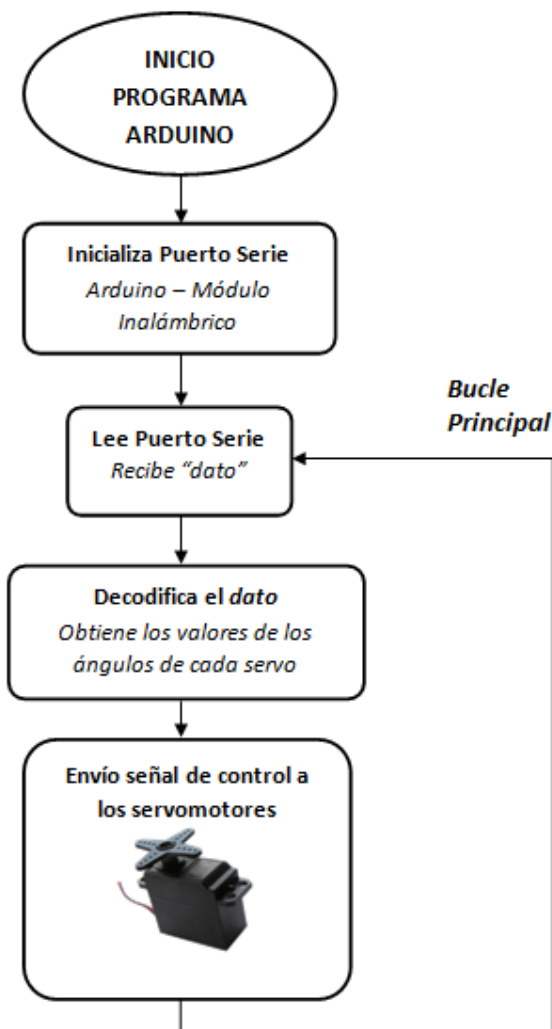
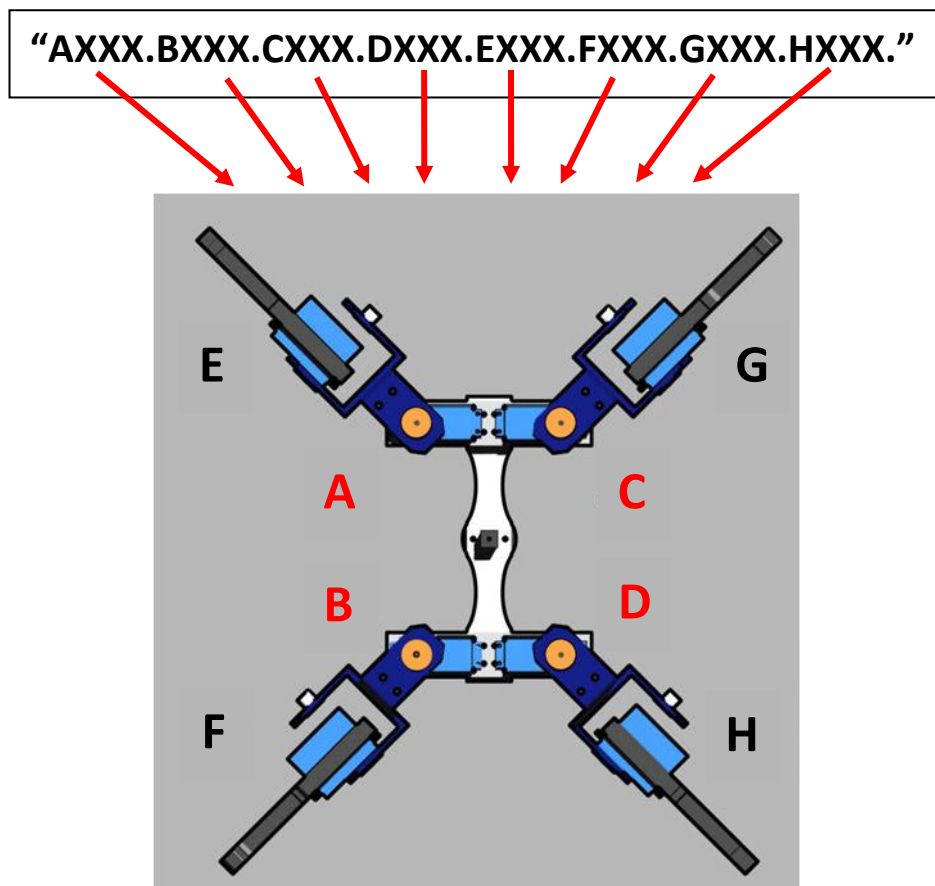


Figura 78 - Esquema general del programa Arduino

En la siguiente *Tabla 5* se muestra la relación entre cada letra y el número de servo. Aunque en un principio se nombraron los servos por números, a la hora de programar se han utilizado letras para que el número de servo (un número entero) no se mezcle con el valor del ángulo (también número entero).

	LETRA	NÚMERO SERVO
CADERAS	A	1
	B	3
	C	5
	D	7
PIES	E	2
	F	4
	G	6
	H	8

Tabla 5 - Letra correspondiente a cada servo



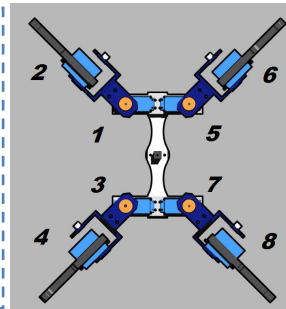
4.3. IMPLEMENTACIÓN FINAL Y EVALUACIÓN DEL AT-AS.

Programación de los movimientos básicos

Aunque en el apartado anterior (4.2. ENVÍO DE DATOS Y CONTROL REMOTO) se ha explicado el funcionamiento de los programas en C++ y en lenguaje Arduino utilizados para el envío y recepción de cadenas de caracteres por *puerto serie*, aún queda mencionar cómo se ha calculado y creado dicho “dato” dentro del programa en C++. A continuación se va a explicar las variables y funciones programadas, necesarias para el cálculo de los ángulos de los servos y que permiten los movimientos básicos del AT-AS.

Las variables enteras que se han utilizado para guardar los valores calculados de los ángulos que han de tener los servos, se han inicializado a 90 de la siguiente forma:

```
int posA = 90; //A - Cadera 1
int posB = 90; //B - Cadera 3
int posC = 90; //C - Cadera 5
int posD = 90; //D - Cadera 7
int posE = 90; //E - Pie 2
int posF = 90; //F - Pie 4
int posG = 90; //G - Pie 6
int posH = 90; //H - Pie 8
```



La cadena de caracteres que se enviará se llama “dato”. Tiene una longitud máxima de 40 caracteres:

```
char dato[39];
```

Como mencionábamos anteriormente, esta cadena va a tener siempre una configuración de la siguiente forma: “AXXX.BXXX.CXXX.DXXX.EXXX.FXXX.GXXX.HXXX.”



La clase “Joystick” hace que el programa C++ detecte qué botones y joysticks del *Gamepad* han sido pulsados. En la Figura 79 se han señalado los botones y Joysticks que se leen con el programa C++ elaborado.

Gracias unas sencillas funciones, podemos guardar en unas variables enteras los valores procedentes del mando.

Figura 79 - Botones y Joysticks del Gamepad

La lectura de los botones que guarda en las variables enteras (*boton1*, *boton2*, *boton3* y *boton4*) un ‘0’ si no se ha pulsado el botón y ‘1’ se ha pulsado, se implementa de la siguiente forma:

```
int boton1 = gpad->getButton(0);
int boton2 = gpad->getButton(1);
int boton3 = gpad->getButton(2);
int boton4 = gpad->getButton(3);
```

La lectura de los *joysticks* que guarda en las variables enteras (*juv1* y *juv2*) valores entre ‘0’ y ‘60’ según la posición vertical de dichos *joysticks*, se ejecuta de la siguiente forma:

```
int juv1 = -((gpad->getAxis(1))*60/32767);
int juv2 = -((gpad->getAxis(3))*60/32767);
```

En la siguiente *Tabla 6* se especifica la acción programada para cada uno de los posibles botones a pulsar:

BOTONES PULSADOS	
Nombre del Botón	Acción programada
<i>boton1</i>	<i>Robot Avanza</i>
<i>boton2</i>	<i>Robot Posición 1</i>
<i>boton3</i>	<i>Robot Posición 2</i>
<i>boton4</i>	<i>Robot Avanza Lento</i>
<i>juv1</i>	<i>Giro derecha</i>
<i>juv2</i>	<i>Giro izquierda</i>

Tabla 6 - Botones pulsados del Gamepad

Las acciones “**Robot Posición 1**” y “**Robot Posición 2**” tan sólo modifican las variables de los ángulos (*posA*, *posB*, *posC*,...etc) una única vez, colocando al robot en dicha posición estática.

La acción “**Robot Avanza**” y “**Robot Avanza Lento**” ejecutan ciclos de movimiento con las cuatro patas de forma que el AT-AS avance en línea recta, aplicando el modo de avance de los cuadrúpedos llamado “alternando par”.

Las acciones "**Giro derecha**" y "**Giro izquierda**" ejecutan los mismos ciclos pero sólo girarán las *caderas* dependiendo de cuánto se ha pulsado los *joysticks*, provocando que el robot gire en una dirección u otra.

Para la programación de los ciclos de avance se han utilizado dos funciones que dependen de unas variables ("*arco_d*" y "*arco_i*") que representan los arcos que giran las *caderas* del AT-AS como ilustra la *Figura 80*.

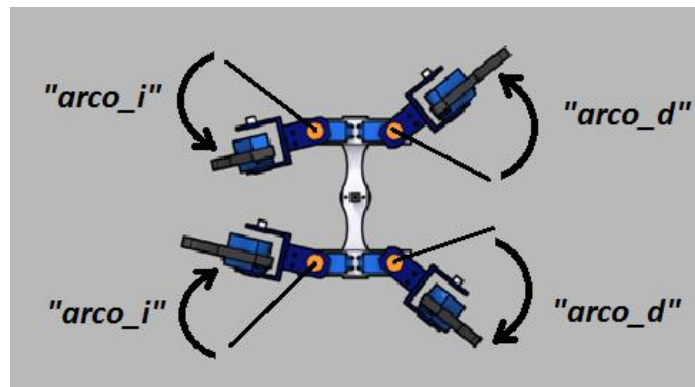


Figura 80 - Arco izquierdo y derecho

En el caso de las acciones de "**Giro derecha**" y "**Giro izquierda**", las dos variables de los arcos toman los valores leídos de los *joysticks* de la siguiente forma:

```
arco_i = jv1;
arco_d = jv2;
```

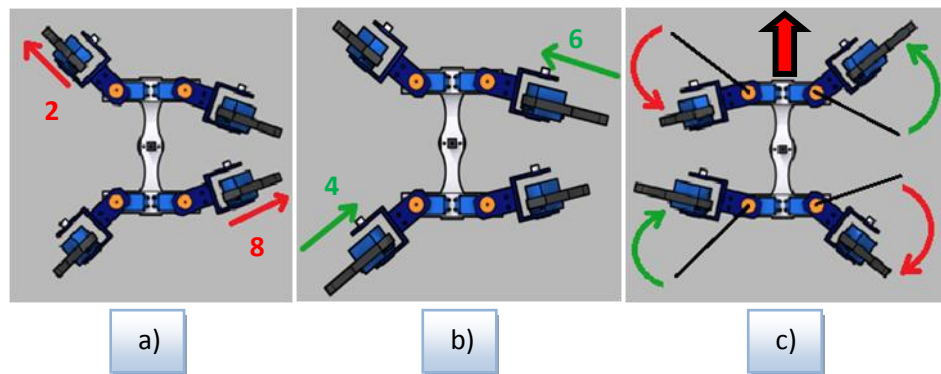
Y en el caso de las acciones de avance en línea recta, los arcos serán máximos y constantes, es decir, de 60 grados:

```
arco_i = 60;
arco_d = 60;
```

A continuación se explican brevemente los dos ciclos que tienen lugar en el avance del AT-AS:

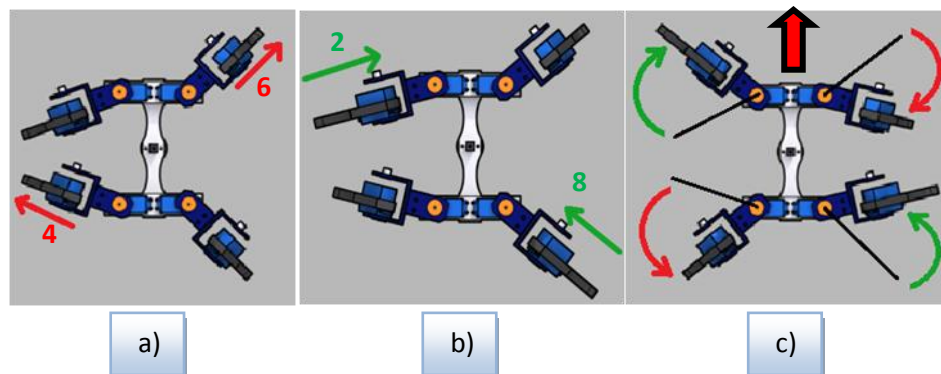
CICLO 1:

- Los pies 2 y 8 bajan tomando apoyo con la superficie del suelo.
- Los pies 4 y 6 suben elevándose para no estar apoyados sobre el suelo.
- Las articulaciones de las caderas describen ángulos correspondientes a los arcos "*arco_i*" y "*arco_d*". De esta forma los *pies 2 y 8* provocan el avance del AT-AS mientras que los *pies 4 y 6* recuperan posición por el aire.



CICLO 2:

- Los pies 4 y 6 bajan tomando apoyo con la superficie del suelo.
- Los pies 2 y 8 suben elevándose para no estar apoyados sobre el suelo.
- Las articulaciones de las caderas describen ángulos correspondientes a los arcos “arco_i” y “arco_d”. De esta forma los pies 4 y 6 apoyados provocan el avance del AT-AS mientras que los pies 2 y 8 recuperan posición por el aire.



Ejecutando estos dos ciclos una y otra vez, el cuadrúpedo es capaz de avanzar y girar su trayectoria de forma eficiente.



Primeras pruebas y cambios

En este apartado de la memoria se exponen las primeras pruebas realizadas con el robot AT-AS y los cambios que dichas pruebas provocaron en el diseño del cuadrúpedo.

A. Gomas antideslizantes

En la primera prueba de locomoción realizada nada más imprimir y montar el conjunto completo del AT-AS, se observó que los *pies* resbalaban demasiado sobre superficies lisas, ya que la parte redonda de plástico apoyada no ofrece suficiente resistencia al avance sobre el suelo. Se decidió por tanto añadir a la pieza un trozo de **goma elástica** de 5mm de ancho y 2mm de grosor. Se cortaron cuatro tiras de 5cm como se aprecian en la *Figura 81*.



Figura 81 - Goma elástica recortada en tira de 5cm

A continuación se pegaron las tiras de goma elástica con pegamento instantáneo al borde redondeado de los *pies* del AT-AS, de forma que quedaron como se ve en la *Figura 82*.

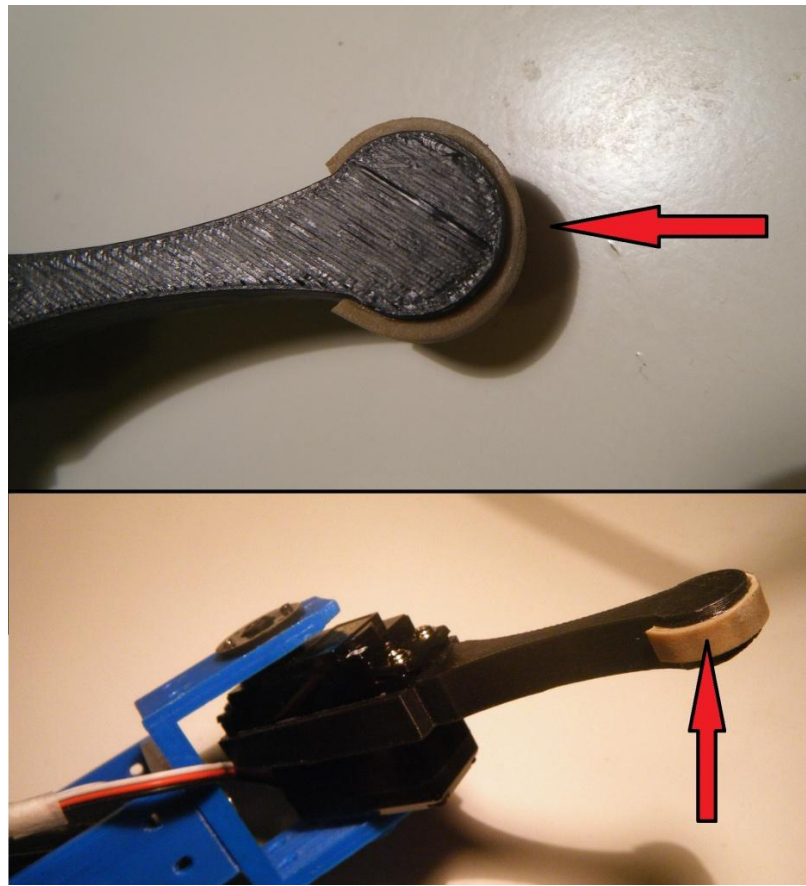


Figura 82 - Detalle de las gomas de los pies

B. Primeras pruebas con el módulo Bluetooth:

Las primeras pruebas de comunicación inalámbrica que se realizaron con el módulo *Bluetooth* no dieron buenos resultados. Aunque en un primer momento el sistema parecía funcionar correctamente (se consiguió enviar y recibir los datos completos), el módulo *Bluetooth* dejaba de recibir y se desconectaba al poco tiempo de su funcionamiento. Esta desconexión podía ser debida a interferencias o a que el módulo se alejaba demasiado del PC. El rango de alcance parecía ser de tan sólo unos pocos metros (entre 0 y 3m) antes de dar problemas de comunicación.

Por este motivo, se decidió probar con el segundo método de comunicación inalámbrica: los módulos *Xbee*, que no han dado problemas de conexión hasta la fecha. Se pueden utilizar ambos métodos inalámbricos para controlar el AT-AS, sin embargo, los módulos *Xbee* dan mejores resultados.

CAPÍTULO 5: CONCLUSIÓN

OBJETIVOS CUMPLIDOS

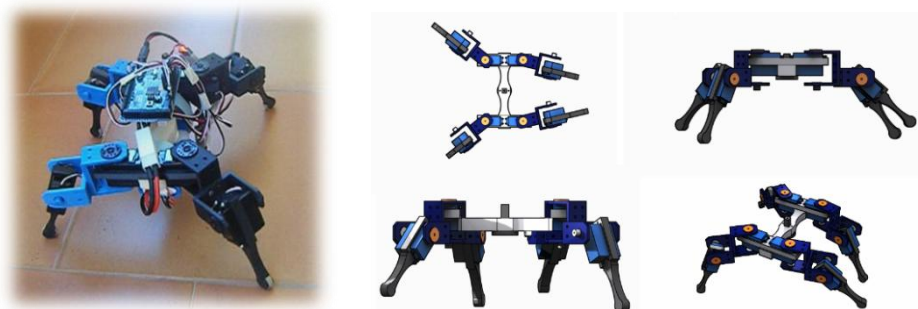
Podemos decir con total certeza que el proyecto *AT-AS* consistente en el diseño, fabricación e implementación de locomoción de un cuadrúpedo imprimible, ha cumplido satisfactoriamente con todos los objetivos marcados desde el inicio de su creación.

Se han analizado los distintos modelos de robots *n-podos*, se ha elegido el modelo físico óptimo para diseñar: un cuadrúpedo de dos grados de libertad por pata. Se han diseñado las piezas que componen el *AT-AS*, se han impreso en impresoras 3D y se ha comprobado que son ligeras pero resistentes. Se ha montado la estructura física y los componentes electrónicos necesarios para su control (incluidos *Gamepad* y módulos inalámbricos). Y por último, evaluando el diseño y programando los movimientos básicos se ha implementado con éxito la locomoción del robot.

En definitiva, se ha conseguido desarrollar un proyecto de **robot eficaz, eficiente y flexible** en comparación con otros proyectos de robots *n-podos*:

- **Robot eficaz y eficiente** ya que con pocos recursos técnicos y económicos se ha conseguido una base útil y completa de robot andador imprimible.
- **Robot flexible** ya que su diseño está abierto a muchas posibilidades de evolución de cara al futuro como se explica en el siguiente apartado “*TRABAJOS FUTUROS*”.

Siguiendo los *link* de *Youtube* que vienen a continuación, se puede ver el resultado en forma audiovisual del proyecto *AT-AS*: el movimiento real del robot y una simulación con las distintas vistas del avance.



Vídeo 1: <http://www.youtube.com/watch?v=ae2UiOCUZeY&feature=g-upl>

Vídeo 2: <http://www.youtube.com/watch?v=OWSHn9gSAzA&feature=plcp>

TRABAJOS FUTUROS

El modelo AT-AS tiene grandes posibilidades de cara al futuro. Es un diseño flexible que acepta considerables cambios que le añadirían ventajas si se adoptasen. Algunos de los posibles cambios se plantean a continuación:

✓ Posibilidad de mutación del modelo cuadrúpedo a un modelo hexápodo:

El diseño modular del AT-AS se ha pensado desde su inicio con medidas y piezas para poder evolucionar a un modelo hexápodo de forma muy sencilla y sin excesivas complicaciones o cambios. Para conseguir tal propósito, tan sólo hay que añadir otras dos patas al centro del cuadrúpedo, acoplándolas al *esqueleto* principal como puede verse en la *Figura 83*.

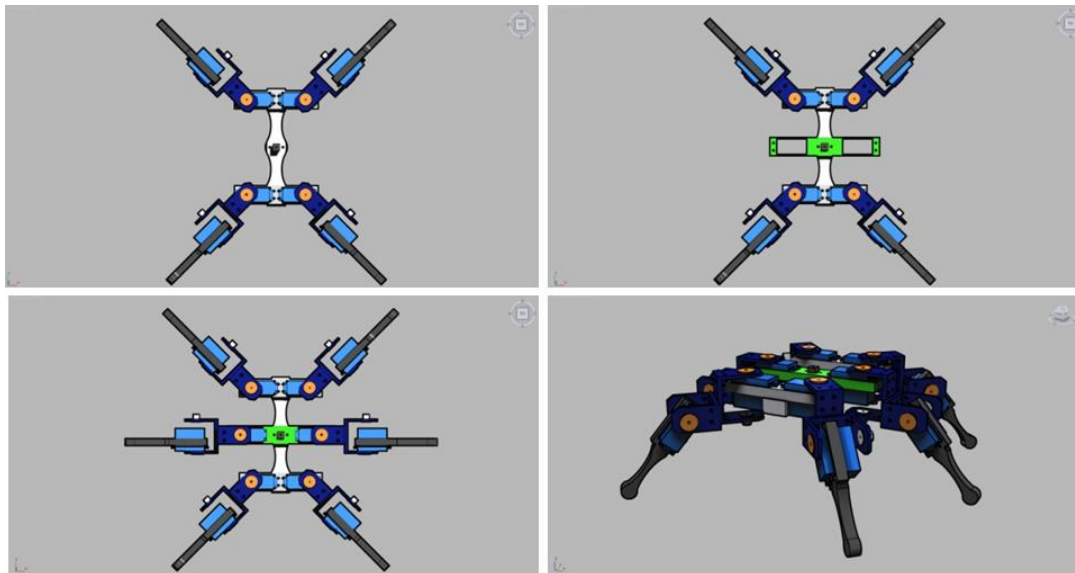


Figura 83 - Evolución a hexápodo del diseño AT-AS

Aunque el robot desarrollado en este Trabajo de Fin de Grado es el modelo cuadrúpedo y no el hexápodo, ya se ha diseñado la nueva pieza ‘*cadera central*’, similar a las piezas llamadas ‘*caderas*’ que sujetan las cuatro patas iniciales a la columna del artrópodo. Esta nueva *cadera central* es más larga y con un orificio cuadrado en el centro para poderla encajar en el *eje cuadrado*. Se puede apreciar dicha pieza marcada con un color verde tanto en la *Figura 83* como en la siguiente *Figura 84*.

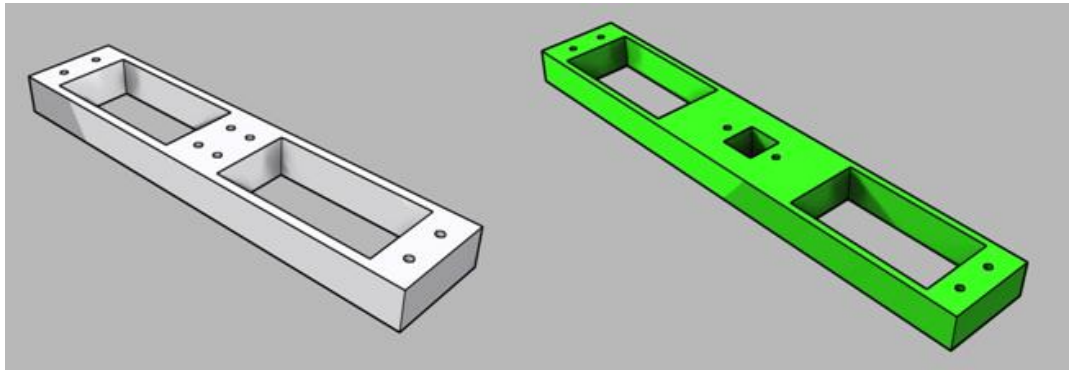


Figura 84 - Diseño de las piezas 'cadera lateral' y 'cadera central'

✓ **Adaptabilidad para invertir el plano de avance:**

Si se diese el caso de vuelco del robot andador AT-AS, tanto si está en forma de cuadrúpedo como de hexápodo, podrían programarse movimientos de avance invertido como se aprecia en la *Figura 85*.

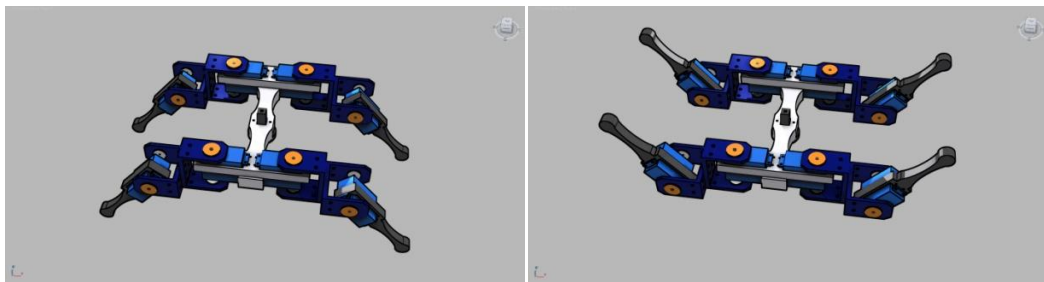


Figura 85 - Adaptabilidad del AT-AS si se invierte el plano de avance

✓ **Adición de sensores para un mejor control:**

Como se ha mencionado en el apartado 2.3. *ARDUINO MEGA 2560*, la cantidad de salidas/entradas analógicas y digitales que tiene la placa *Arduino Mega 2560*, permiten la utilización de sensores de infrarrojos o sensores de ultrasonidos para detectar obstáculos y también de sensores de presión en las terminaciones de las patas para mejorar el control de la locomoción ante desniveles e irregularidades del terreno.

✓ **Otras ventajas y evoluciones del AT-AS:**

Otras ventajas del AT-AS son la posibilidad de mejorar el diseño mecánico de las patas y la estructura modular de manera que la modificación de cualquier eslabón y pieza se realice sin dificultades, la flexibilidad en el diseño electrónico que permite generar plataformas estándar de control y programación de más movimientos, y por último, la comunicación con otros robots por medio de redes de protocolo ZigBee (redes 'mesh' gracias a los módulos Xbee).

A TÍTULO PERSONAL

Esta experiencia me ha servido para ampliar mis conocimientos ingenieriles y poner a prueba mis habilidades como desarrolladora de proyectos de robótica.

Durante el desarrollo de este Trabajo de Fin de Grado me enfrenté a muchos problemas inesperados, los cuales he conseguido superar gracias a mi pasión por la robótica y mi esfuerzo por lograr todos los objetivos que me había planteado desde el principio.

Ahora sé lo que es estar involucrada en un proyecto grande y complejo, fijar unas metas y llegar a alcanzarlas para así poder seguir avanzando hacia nuevos objetivos que se abren de cara al futuro.

Además, ayudar a la expansión de la base de datos de piezas imprimibles de la web *Thingiverse* evolucionando piezas diseñadas por otros usuarios y añadiendo nuevas aportaciones es algo en lo que personalmente me gustaría seguir contribuyendo, porque con pequeñas aportaciones como éstas se enriquecen los conocimientos de la humanidad.

CAPÍTULO 6: BIBLIOGRAFÍA Y REFERENCIAS

1. Blog Ingenieria y Computacion. [En línea] [Consultado el: 28 de Agosto de 2012.] <http://ingenieriaycomputacion.blogspot.com.es/2010/03/timberjack-walking-machine-vehiculo.html>.
2. Artículo de Newscientist sobre el robot ATHLETE. [En línea] [Consultado el: 10 de Agosto de 2012.] <http://www.newscientist.com/article/dn13590>.
3. Object Oriented Mechanics Library. [En línea] [Consultado el: 15 de Agosto de 2012.] <http://iearobotics.com/oowlwiki/doku.php>.
4. Web de la Asociación de Robótica de la Carlos III de Madrid. [En línea] [Consultado el: 31 de Agosto de 2012.] http://asrob.uc3m.es/index.php/Proyecto:_Clone_wars.
5. Librería "servo.h" de Arduino. [En línea] [Consultado el: 6 de Julio de 2012.] <http://arduino.cc/es/Reference/Servo>.
6. Web del proyecto 'Asterisk', hexápodo de la Universidad de Osaka. [En línea] [Consultado el: 10 de Julio de 2012.] http://www-arailab.sys.es.osaka-u.ac.jp/research/limbgroup/e_index.html.
7. Rex Bionics. [En línea] [Consultado el: Julio 10, 2012.] <http://www.rexbionics.com/>.
8. Wikipedia. *Denavit Hartenberg parameters*. [En línea] [Consultado el: Agosto 26, 2012.] http://en.wikipedia.org/wiki/Denavit%E2%80%93Hartenberg_parameters.
9. "bryanandaimee". Thingiverse - Versatile Servo Bracket System. [En línea] [Consultado el: 22 de Mayo de 2012.] <http://www.thingiverse.com/thing:5784>.

ANEXOS

ANEXO I: PRESUPUESTO Y TIEMPO EMPLEADO

TIEMPO EMPLEADO:

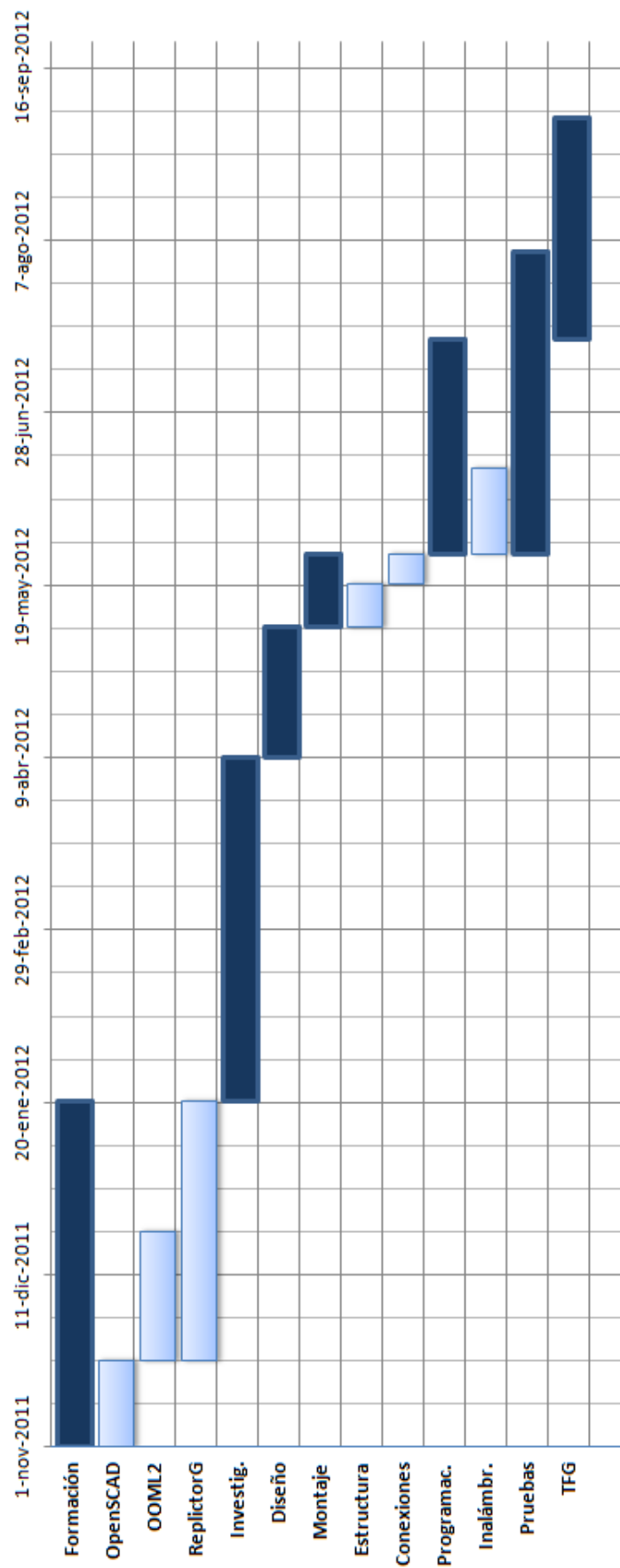
El tiempo total empleado en la realización del proyecto AT-AS ha sido de un total de 10 meses, 2 de los cuales han sido usados para el aprendizaje de las materias y programas necesarios para la impresión 3D y los 8 restantes para la investigación y diseño del propio AT-AS.

A continuación se expone una tabla con las actividades que han tenido lugar a lo largo de esos 10 meses de proyecto, con las fechas de inicio y finalización de las mismas.

ACTIVIDAD	NOMBRE	DURACIÓN	ACTIVIDAD PRECEDENTE	INICIO	FINALIZACIÓN
A	Formación	80	-	01/11/2011	19/01/2012
B	OpenSCAD	20	-	01/11/2011	20/11/2011
C	OOML2	30	B	21/11/2011	20/12/2011
D	ReplicatorG	60	B	21/11/2011	19/01/2012
E	Investigación	80	A	20/01/2012	08/04/2012
F	Diseño AT-AS	30	E	09/04/2012	08/05/2012
G	Montaje AT-AS	17	F	09/05/2012	25/05/2012
H	Estructura	10	F	09/05/2012	18/05/2012
I	Conexiones	7	H	19/05/2012	25/05/2012
J	Programación	50	G	26/05/2012	14/07/2012
K	Inalámbrico	20	G	26/05/2012	14/06/2012
L	Pruebas	70	G	26/05/2012	03/08/2012
M	TFG	51	J	15/07/2012	03/09/2012

En la siguiente *página 83* puede verse el *Diagrama de Gantt* correspondiente al tiempo empleado en cada actividad.

DIAGRAMA DE GANTT



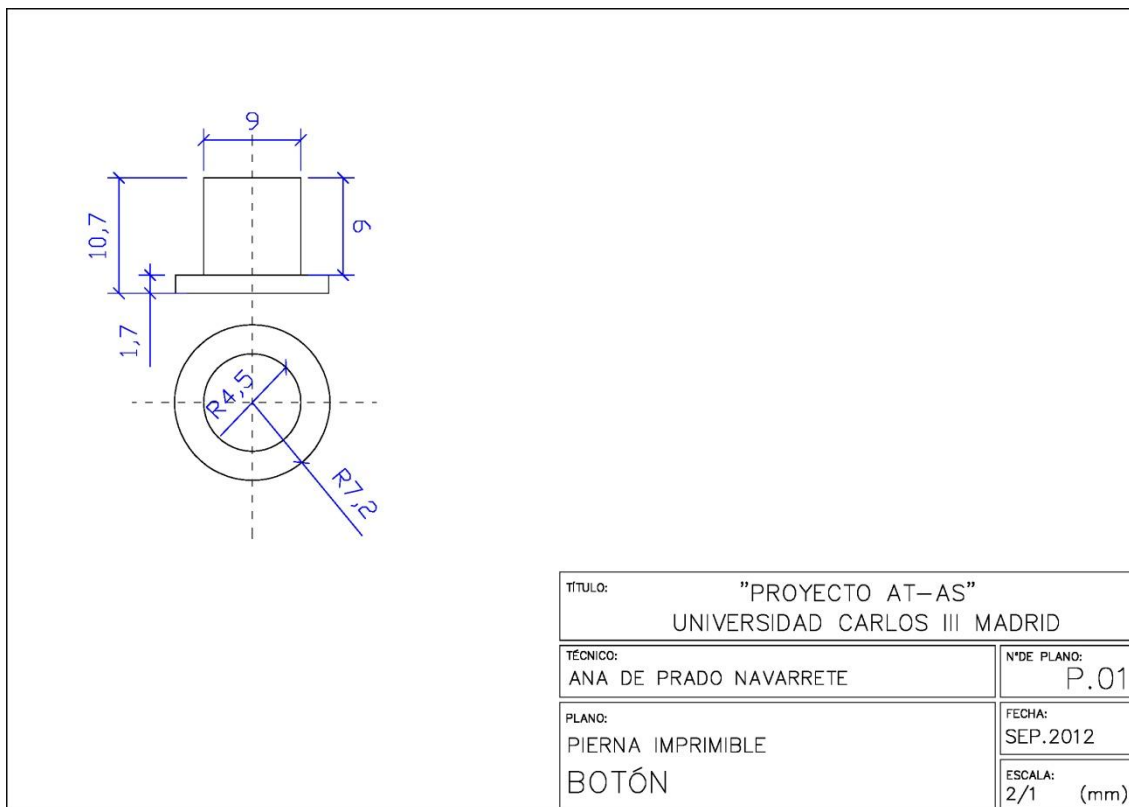
PRESUPUESTO DEL PROYECTO AT-AS:

CÓDIGO	DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	PRECIO TOTAL
01.	BLOQUE 1 - Software			0,00 €
01.01	<i>OpenSCAD</i>	1	0,00 €	0,00 €
01.02	<i>Qt Creator y Biblioteca OOML</i>	1	0,00 €	0,00 €
01.03	<i>ReplicatorG</i>	1	0,00 €	0,00 €
01.04	<i>Arduino 0019</i>	1	0,00 €	0,00 €
02.	BLOQUE 2 - Actuadores			32,00 €
02.01	<i>Servomotor Futaba S3003</i>	8	4,00 €	32,00 €
03.	BLOQUE 3 - Microcontrolador			39,00 €
03.01	<i>Arduino Mega 2560</i>	1	39,00 €	39,00 €
04.	BLOQUE 4 - Alimentación			37,20 €
04.01	<i>LiPo DragonRed (2S, 5000mAh y 30C)</i>	1	21,30 €	21,30 €
04.02	<i>Cargador LiPo RC imax B3 (2-3S)</i>	1	12,50 €	12,50 €
04.03	<i>UBEC Hobbywing (5-6V, max 5A)</i>	1	3,40 €	3,40 €
05.	BLOQUE 5 – Control Remoto			106,50 €
05.01	<i>Gamepad Thrustmaster FireStorm</i>	1	13,50 €	13,50 €
05.02	<i>Módulo Bluetooth</i>	1	12,50 €	12,50 €
05.03	<i>Xbee Zigbee Shield</i>	1	14,20 €	14,20 €
05.04	<i>Módulo Xbee (S1, 1mW, antena)</i>	2	23,20 €	46,40 €
05.05	<i>Xbee Explorer USB</i>	1	19,90 €	19,90 €
07.	BLOQUE 7 – Herramientas y otros elementos			11,00 €
07.01	<i>Herramientas y otros materiales</i>	-	11,00	11,00 €
TOTAL PROYECTO				225,70 €

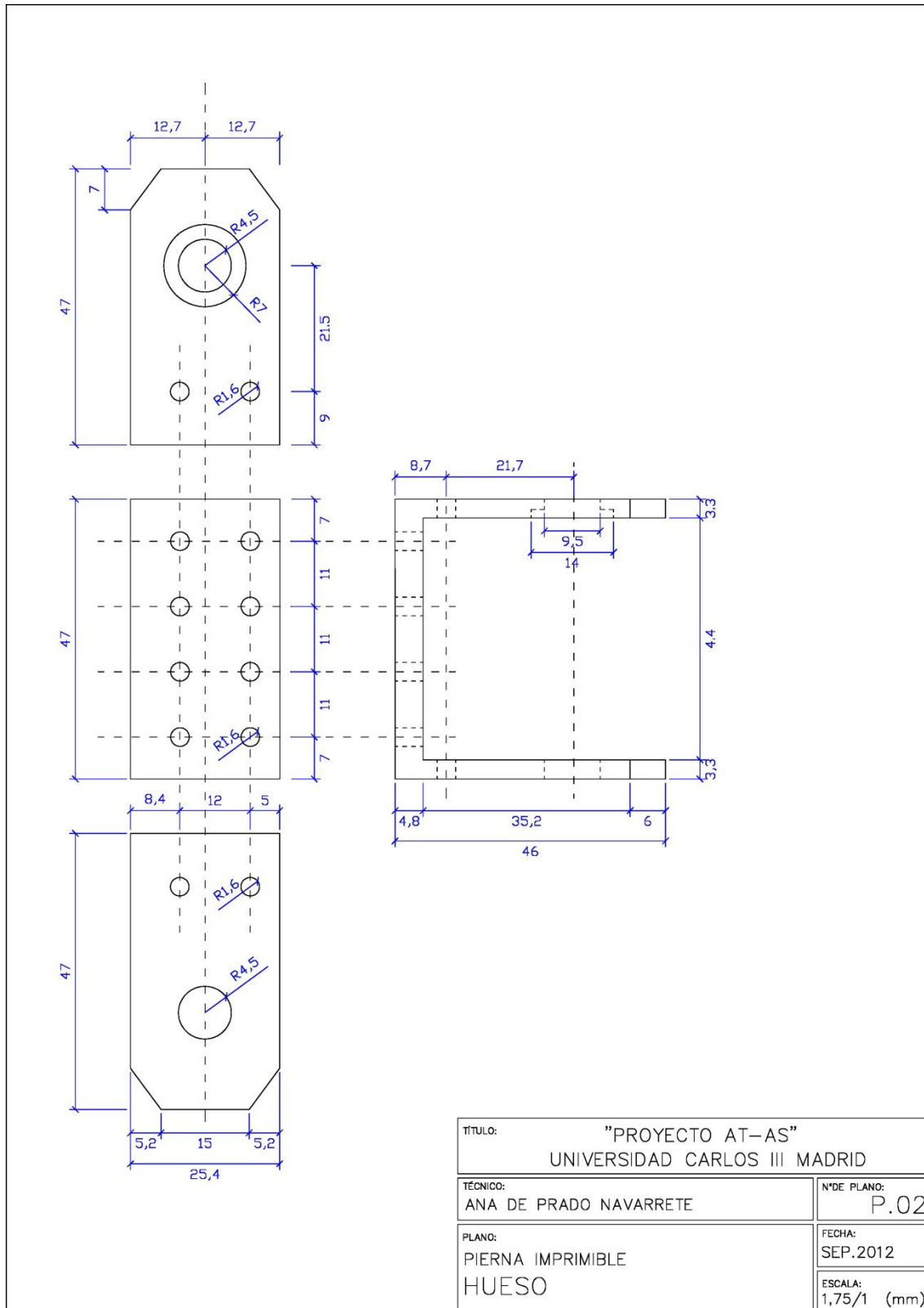
El presupuesto total del proyecto asciende a **225,70 €**. Sin embargo, hay algunas posibilidades de fabricar e implementar el AT-AS abaratando costes. Como vemos en la tabla anterior, el bloque más caro es el *Bloque 5 – Control Remoto*, que asciende a 106,50 €. Si suprimiésemos el modo de comunicación inalámbrica *Xbee* (señalada en rojo) y utilizásemos tan sólo el módulo *Bluetooth*, se reduciría el precio del proyecto hasta quedar un presupuesto total de **145,20 €**.

ANEXO II: PLANOS DE LAS PIEZAS DISEÑADAS

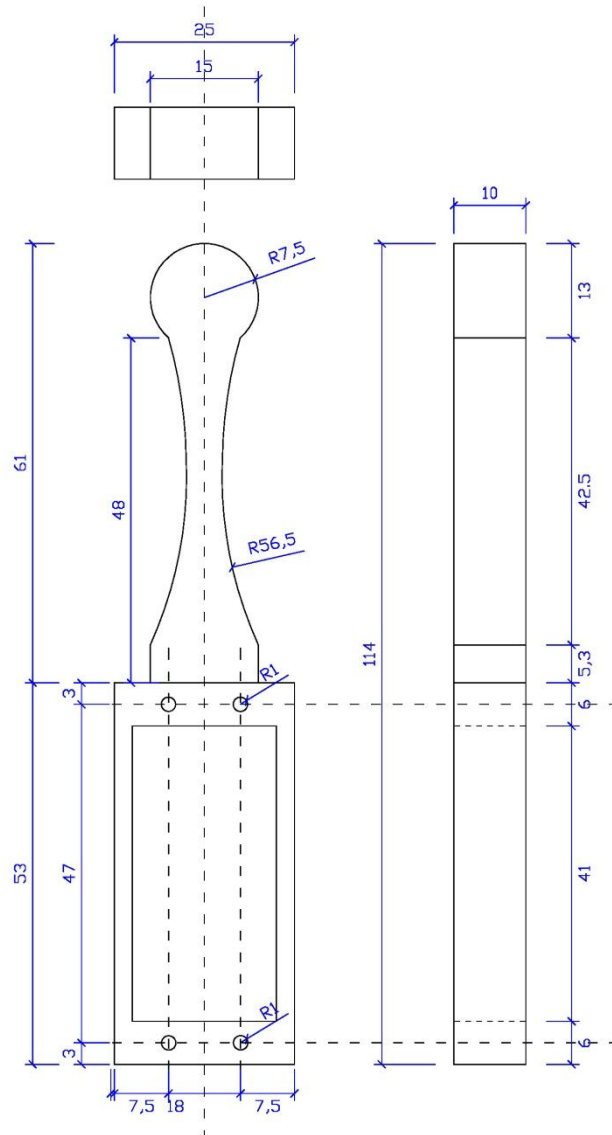
P.01: PIERNA IMPRIMIBLE - PIEZA BOTÓN



P.02: PIERNA IMPRIMIBLE - PIEZA HUESO

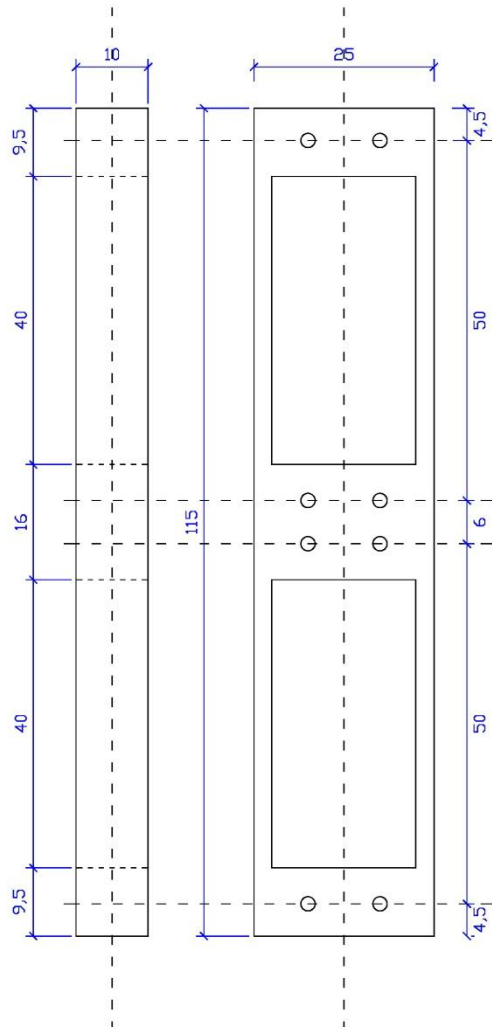


P.03: PIERNA IMPRIMIBLE - PIEZA PIE



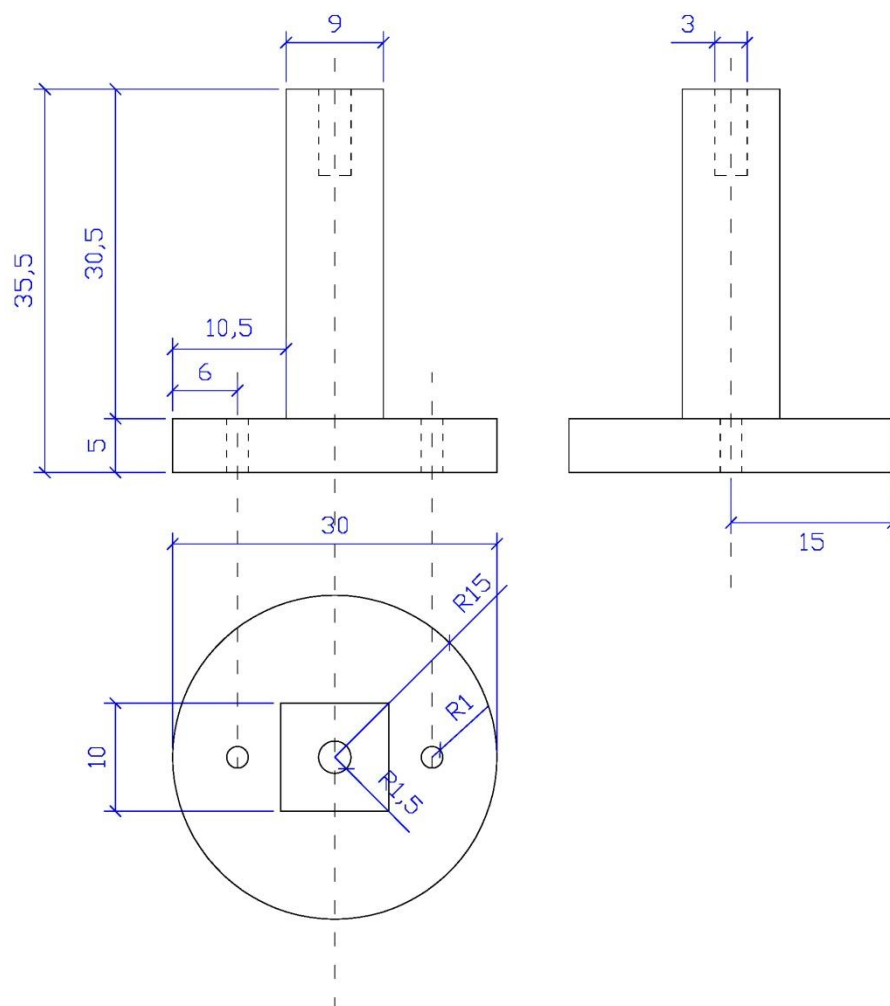
TÍTULO: "PROYECTO AT-AS" UNIVERSIDAD CARLOS III MADRID	
TÉCNICO: ANA DE PRADO NAVARRETE	Nº DE PLANO: P.03
PLANO: PIERNA IMPRIMIBLE PIE	FECHA: SEP.2012
	ESCALA: 1,75/1 (mm)

E.01: ESQUELETO IMPRIMIBLE - PIEZA CADERA



TÍTULO:		"PROYECTO AT-AS"	
		UNIVERSIDAD CARLOS III MADRID	
TÉCNICO:	ANA DE PRADO NAVARRETE	Nº DE PLANO:	E.01
PLANO:	ESQUELETO IMPRIMIBLE CADERA	FECHA:	SEP.2012
		ESCALA:	1,75/1 (mm)

E.03: ESQUELETO IMPRIMIBLE - PIEZA EJE CUADRADO



TÍTULO: "PROYECTO AT-AS" UNIVERSIDAD CARLOS III MADRID		
TÉCNICO: ANA DE PRADO NAVARRETE		Nº DE PLANO: E.03
PLANO: ESQUELETO IMPRIMIBLE EJE CUADRADO		FECHA: SEP.2012
		ESCALA: 2/1 (mm)

E.04: EVOLUCIÓN ESQUELETO IMPRIMIBLE - PIEZA CADERA CENTRAL

